Integrating Disambiguation and User Preferences into Large Language Models for Robot Motion Planning

Mohammed Abugurain and Shinkyu Park

Abstract— **This paper presents a framework that can interpret humans' navigation commands containing temporal elements and directly translate their natural language instructions into robot motion planning. Central to our framework is utilizing Large Language Models (LLMs). To enhance the reliability of LLMs in the framework and improve user experience, we propose methods to resolve the ambiguity in natural language instructions and capture user preferences. The process begins with an ambiguity classifier, identifying potential uncertainties in the instructions. Ambiguous statements trigger a GPT-4-based mechanism that generates clarifying questions, incorporating user responses for disambiguation. Also, the framework assesses and records user preferences for non-ambiguous instructions, enhancing future interactions. The last part of this process is the translation of disambiguated instructions into a robot motion plan using Linear Temporal Logic. This paper details the development of this framework and the evaluation of its performance in various test scenarios.**

I. INTRODUCTION

Recent advancements in Large Language Models (LLMs) [1]–[7] demonstrate their remarkable capabilities, exemplified by their ability to process and generate human-like text. These advanced computational models have revolutionized various domains, including robotics, where LLMs are increasingly employed to bridge the communication gap between humans and machines. This allows robots to execute tasks based on complex human instructions. However, as LLMs are widely used in robotics applications, an increasing number of challenges arise in accurately interpreting humans' natural language commands that are ambiguous or contain their personal preferences, which are unknown to the robots.

Our research is motivated by growing concerns about *hallucination* and *inconsistency* in LLMs as reported in recent literature [8]–[10]. Hallucination refers to generating false or misleading information that doesn't accurately reflect the input provided. This is particularly problematic in robotic applications, as LLMs can misinterpret humans' instructions or generate unsafe commands when they hallucinate, ultimately impeding the applicability of LLM-integrated systems. Additionally, the inconsistency, referring to the phenomenon of LLMs producing different responses to the same query,

further complicates the use of LLMs in scenarios where reliability of the robotic systems is crucial. These issues limit the functional scope of robots and raise concerns about safety, especially in environments where human-robot interaction is frequent. In this work, we address these limitations by enhancing the reliability of LLMs for robot motion planning and execution. We propose a framework that effectively detects and corrects misinterpretation and inconsistencies, thereby bridging the gap between the potential of LLMs and their practical, reliable application in robotics.

Our framework, which can be integrated into existing robotic systems using LLMs, minimizes ambiguity in human natural language instructions, enabling robots to understand and execute the instructions with improved reliability. This is challenging because human natural language is inherently ambiguous, where meanings are frequently implied rather than explicitly stated. For instance, given an instruction *"Go to the room where I forgot my keys"*, the robot must understand where the user forgot their keys to plan and execute the navigation command. Furthermore, the robot needs to determine whether a given human instruction contains all the information about their preferences required to execute the instructed task. To illustrate, consider an instruction *"Go to the room I like the most"*. The instruction appears unambiguous as there is exactly one room the user likes the most, but it does not indicate which room they like the most. In this example, the robot must detect that the instruction is incomplete and ask the user for more information about their preference.

Drawing from these two examples, we consider ambiguity as the uncertainty in a sentence's meaning that allows for multiple, equally probable interpretations. Instructions with unspecified preferences, though not ambiguous, are regarded as incomplete, necessitating additional information on the preferences to complete the task. This paper posits that ambiguity is a universal phenomenon, not specific to any particular user, while preference is user-specific.

As our main contributions, we develop a framework capable of resolving ambiguity in natural language instructions and clarifying them through meaningful queries. Additionally, it can detect whether the instructions contain all the necessary information regarding user preferences, query the users whenever additional details are required to complete the task, and also retain this information for future use. The rest of the paper is organized as follows: We present the background and related work in Section II, and the problem formulation and proposed framework in Section III. We validate the efficacy of our framework through experiments

in Section IV, and conclude the paper with summaries and discussions in Section V.

II. Related Work

a) Instructing Robots by Natural Language: This is appealing because the end user does not need to learn technical skills to operate the robot, ensuring smoother and more accessible experience. The work of [11] integrates free-form natural language into imitation learning and robot perception into a single neural network model. This approach enabled users to instruct robots on their manipulation tasks using natural language. The work of [12] introduces a probabilistic framework to instruct robots to navigate unknown environments using natural language instructions. With advancements in LLMs, recent works have utilized LLMs to translate human natural language into robot commands [13]–[17]. The work of [13] provides a Python-based interface to integrate a LLM, allowing a robot to retrieve information about its surrounding environment and manipulate objects. Based on natural language instructions from users, the LLM generates a sequence of Python functions for the robot to execute in order to complete the required object manipulation task. The work of [14] presents a natural language-based robot planning framework for multi-step sequential manipulation tasks. This framework constructs task and policy plans to achieve symbolic goals as instructed by a user.

b) Resolving Ambiguity in Natural Language: A substantial body of recent work investigates methods to reduce the ambiguity in natural language instructions for robotics applications. To mention a few, the work of [18] utilizes LLMs for evaluating the clarity of user commands for human-robot interactions, with the goal of enhancing the reliability of robot operations by minimizing unintended actions. Using LLMs with with the map of the environment, they introduce a method to classify commands as clear, ambiguous, or infeasible. For ambiguous commands, a question-generation technique is employed to seek clarification.

The work of [19] develops a framework designed to enhance the efficacy of LLMs in multi-step planning and commonsense reasoning within robotics applications. This framework improves the LLMs' ability to recognize their own uncertainty, prompting them to seek human inputs as necessary. It incorporates conformal prediction theory to provide statistical guarantees for a robot's completion of instructed tasks with minimal human intervention. Similarly, the work of [20] introduces a framework that enables robots to request clarifications in conversations with human users regarding ambiguous object identifications. This approach aims to improve efficiency and user satisfaction by leveraging the knowledge about the robot's surrounding environment to reduce the frequency of such requests, thereby minimizing the need for human intervention.

Though we share a similar motivation with previous studies [18]–[20] to disambiguate human instructions, our work stands apart by focusing on robot navigation for which we design a framework that directly detects ambiguity in the instructions. The framework is required to translate human natural language instructions into LTL specifications, necessitating the development of a tailored framework. Consequently, as shown in Table I, our framework outperforms the method of [18] in terms of ambiguity detection accuracy within the targeted applications. Additionally, our framework integrates the retention of information about user preferences through a proper querying process, which can be utilized in future interactions.

c) User Preference Identification and Retention: This is a crucial step toward enabling robots to understand and adapt to individual user preferences. Such adaptability is essential for enhancing the user experience and building trust between humans and robots. The work of [21] proposes a decision-theoretic model based on a POMDP formulation to improve human-robot interaction. This approach enables robots to learn user preferences in real time, despite ambiguity and noise in their speech, thereby reducing the need for offline model training. The model's effectiveness was demonstrated through simulations and experiments in a robotic wheelchair application. More recent work [22] focuses on implementing a robot-assisted household cleanup system in which LLMs for few-shot summarization are utilized to learn user preferences from a few examples. These preferences include the designated places for certain household objects.

III. Problem Formulation and Methods

Consider an environment containing m locations of interest, denoted as $L = \{l_1, l_2, \ldots, l_m\}$, each assigned a unique label recognizable by a robot, as illustrated in Fig. 4. We investigate the problem of planning a path $P = \{p_1, p_2, \ldots, p_n\}$ for robot navigation. The start point $p_1 \in L$, all the intermediate locations $\{p_2, \cdots, p_{n-1}\} \subset L$, and the goal point $p_n \in L$ are to be selected according to human instructions provided in natural language, where we assume that the instructions can be specified by LTL. When the user instructions are clear, existing approaches allow for translating natural language instructions into LTL specifications [23]–[25] and facilitate robot motion planning with LTL specifications [26]. Therefore, we aim to develop a framework that removes ambiguity and identifies user preferences in the instructions. Fig. 3 illustrates a dialog between a user and a robot during the disambiguation and user preference identification process within our framework. A robot navigation path resulting from the dialog is illustrated in Fig. 4.

We utilize the `text-embedding-ada-002` model [27] and a random forest classifier to analyze ambiguity in the instructions. `text-embedding-ada-002` is a LLM trained on a large text corpus. It analyzes and converts textual input into high-dimensional vectors, capturing intricate semantic and contextual nuances, which can be used for various natural language processing applications, including text classification [28]. In our framework, the output of `text-embedding-ada-002` is fed into a trained random forest classifier to determine whether a sentence is ambiguous. The random forest model is beneficial as it can classify
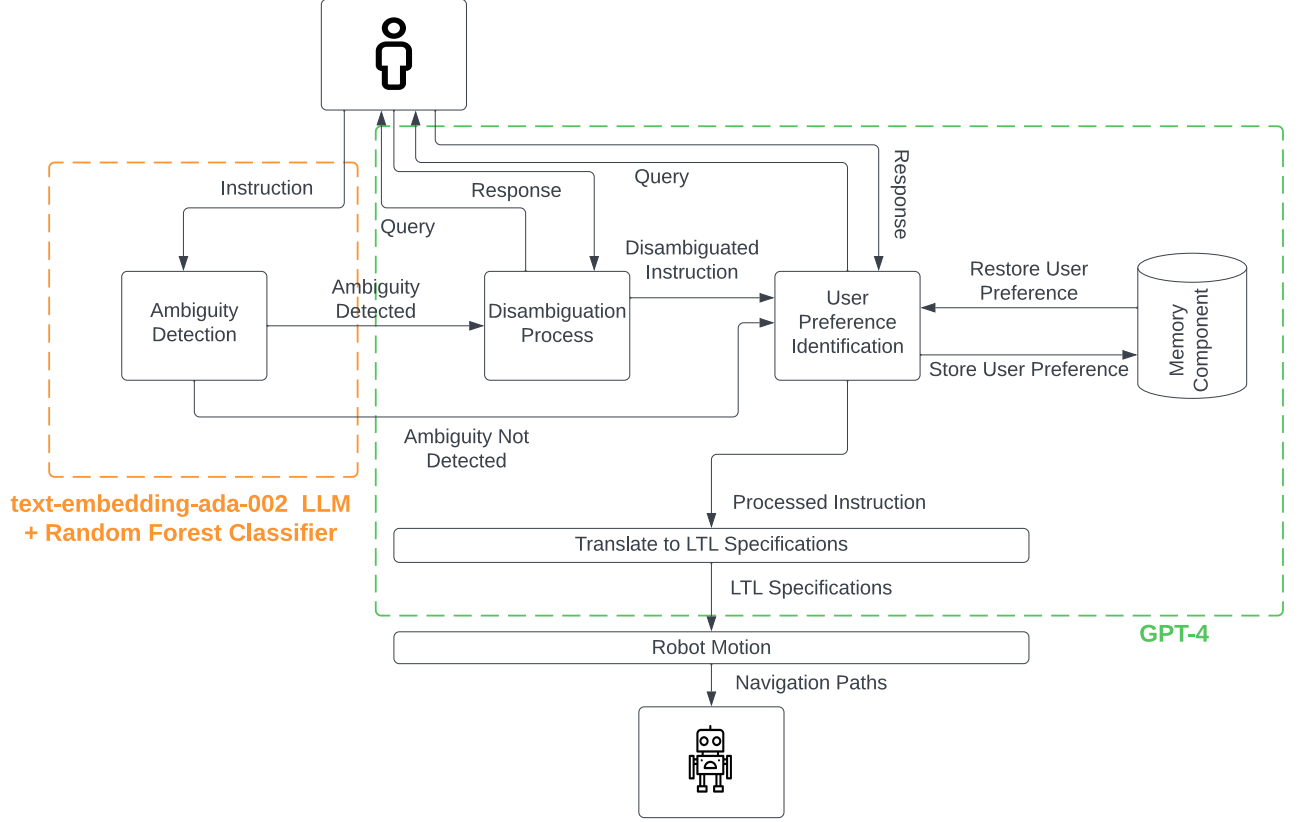
Fig. 1: Our proposed framework consisting of `text-embedding-ada-002` with a random forest classifier for the ambiguity detection in human instructions, and GPT-4 for the disambiguation process and user preference identification. The framework integrates existing methods for translating the processed instructions into LTL specifications and for planning navigation paths based on these specifications.

the vector data produced by `text-embedding-ada-002` without the need for dimensionality reduction.

When ambiguity is detected in the instructions, our framework employs the GPT-4 model to generate contextually relevant questions to resolve it. These questions aim to solicit additional information from the user, effectively narrowing down multiple interpretations of the instructions. The same GPT-4 model then processes the user's responses to produce disambiguated, explicit instructions. Upon removing ambiguity from the original instructions, the framework attempts to detect any unspecified user preferences using GPT-4. We equipped the GPT-4 model with a conversation buffer memory to record users' preferences for future interactions. If the preference is known from previous interactions stored in the memory component, the model will not ask the user for their preference again. Hence, this step is crucial for personalized interactions and enhancing the user experience.

The final step involves translating the processed instructions into LTL specifications. LTL is a formalism used to specify a sequence of events alongside their temporal interrelations. Such specification is crucial in accurately representing scenarios where the order in which a robot's

actions are executed is critical, such as in robotic task planning and navigation. LTL's core strength emanates from its ability to represent complex temporal scenarios within a structured, mathematically rigorous framework. This capability facilitates the precise interpretation and execution of sequences by machines, ensuring high levels of accuracy and predictability. To leverage these compelling attributes, we incorporated LTL representation into the design of our framework.

We adopt the framework developed in [23]–[25] to translate the user instructions into LTL specifications. We have integrated this framework into our framework based on GPT-4. Then, the framework computes paths for robot navigation achieving the LTL specifications, using the approach proposed in [26]. This approach represents the path planning problem as searching for the shortest path over a graph of convex sets, which can be solved using convex optimization.

Through this multifaceted approach, our system not only addresses the challenges posed by the ambiguity of natural language but also enriches the interaction between humans and robots by understanding and adapting to individual user preferences. Fig. 1 illustrates our framework.

IV. EXPERIMENTS

A. Dataset and Training

To optimize the performance of the framework, we train the random forest classifier. We begin with creating a dataset that comprises of 155 ambiguous statements of navigation commands. Each statement is accompanied by a map of the environment, a clarification question, possible valid user responses, and a disambiguated sentence for each valid response. The dataset is generated through the following process: Initially, we generate 20 ambiguous statements of navigation commands across three different environments. Then, we use GPT-4 to paraphrase and manipulate the word order in these statements to produce additional 135 ambiguous statements.[1] Then, we combined our dataset with existing ones from the previous works [29], [30]. The dataset from [29] consists of 3382 commands corresponding to 39 LTL statements for mobile robot manipulation and pick-and-place tasks.[2] Since the dataset is unbalanced, with the number of ambiguous statements being less than that of non-ambiguous statements, we performed upsampling to balance the dataset. Subsequently, 20 ambiguous and 20 unambiguous statements were selected for testing.

Using the dataset, we train the random forest classifier with the standard training configuration of 100 trees in the forest. The input to the classifier is the vector representations of human instructions produced by `text-embedding-ada-002`. The classifier then outputs 0 for non-ambiguous and 1 for ambiguous statements.

B. Ambiguity Detection and Disambiguation Process

The first step in our framework is to analyze the ambiguity of input sentences and classify the sentences as ambiguous or non-ambiguous. We compared the performance of our ambiguity detection model, which consists of `text-embedding-ada-002` combined with a random forest classifier, against that of a range of existing models that could be applied to our problem. These models include `text-embedding-ada-002` with a support vector machine, zero-shot GPT-3.5, zero-shot GPT-4, few-shot GPT-3.5, few-shot GPT-4, BERT, and the CLARA framework proposed in [18]. As summarized in Table I, the `text-embedding-ada-002` model with a random forest classifier outperformed the others in terms of accuracy.

When the input sentence is classified as ambiguous, the framework generates questions to resolve the ambiguity. The questions are generated using GPT-4 by providing the input sentence and the environment map. The questions are then presented to the user, and their responses are fed into GPT-4 to generate a resulting disambiguated sentence. Table II

TABLE I: Performance Comparison of Various LLM-based Ambiguity Detection Methods

Model	Accuracy
ADA + Random Forest	**0.85**
Few-shot GPT-4	0.79
BERT	0.71
Zero-shot GPT-4	0.69
Few-shot GPT-3	0.61
CLARA [18]	0.53
ADA + Support Vector Machine	0.51
Zero-shot GPT-3	0.48

lists a few examples of original ambiguous sentences, clarifying questions, user responses, and resulting disambiguated sentences.

To assess the performance of the disambiguation process, we conducted a series of tests using the dataset we created. These tests involved a comparative analysis of the likelihood of sentences being ambiguous before and after the disambiguation process. Initially, sentences were passed through the ambiguity classifier to determine their likelihood of being ambiguous. Then, ambiguous sentences were refined in our disambiguation process based on user responses. The refined sentences were then re-analyzed by the classifier to assess any changes in their ambiguity likelihood. This approach allowed us to quantitatively measure the effectiveness of the disambiguation process. The results are shown in the bar graph in Fig. 2, which illustrates that the disambiguation process effectively reduces the likelihood of ambiguity by 36.8% on average.

C. Preference Analysis

The next step in our framework involves preference analysis using the GPT-4 model. For this purpose, we design a prompt for the model to process the input sentence, the environment map, and the previous interactions from the memory component. When a user preference is detected in the input sentence, the model will first check its memory component to see if the user preference is already known. If not, the model will generate a question to ask the user for their preference. The user's responses are then fed into the model to generate a sentence that reflects the user's preference. Finally, the model records the user's preference in its memory component.

D. Application to Robot Planning

The dialog depicted in Fig. 3 provides an example of how our framework can be used to translate user navigation instructions into their corresponding LTL specifications. In our example, the user provides an ambiguous instruction, which includes an unspecified personal preference, for a robot to sequentially visit multiple locations in the environment. When the dialog is initiated between the robot and the user,

TABLE II: Examples of the Disambiguation Process

| Ambiguous Sentence	Clarification Question	User Response	Disambiguated Sentence
Go to the room and then pass by the red room	Which room are you referring to when you say "go to the room"?	the green room	Go to the green room and then pass by the red room.
Go to the room and then pass by the red room	Which room should I go to first before passing by the red room?	the blue room	Go to the blue room first and then pass by the red room.
Make your way to the room that feels less crowded	Which room are you referring to as the one that feels less crowded?	the red room	Make your way to the red room that feels less crowded.
Go to the room where we had the meeting.	Which room are you referring to when you mentioned "the room where we had the meeting"?	green room	Go to the green room where we had the meeting.
Head to the room, and then go past the crimson chamber.	Which room are you referring to when you say "the room"? Additionally, could you confirm if the "crimson chamber" refers to the 'red room' on the map?	the room is the blue, and the crimson is the green one	Head to the blue room, and then go past the green chamber.
Once you've bypassed the red room, navigate through the blue room to find your way to the illuminated space	To clarify your command, could you please specify what you mean by the "illuminated space"?	red room	Once you've bypassed the red room, navigate through the blue room to find your way to the red room.
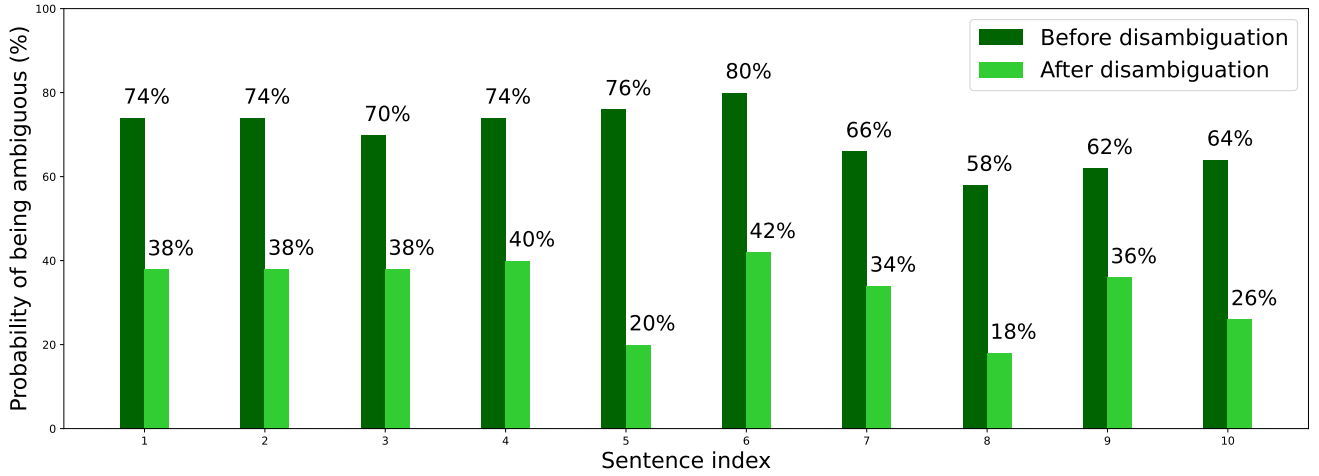
Fig. 2: Ambiguity likelihood before and after the disambiguation process

the framework disambiguates the instruction and identifies the user's preference, and then it translates the processed instruction into the corresponding LTL specifications. We adopt the algorithm in [26] to compute paths for the robot navigation based on the specifications. The robot's navigation along the computed paths is illustrated in Fig. 4 where we can verify that the robot is able to precisely carry out the user's instructions. This verifies the efficacy of our framework.

V. CONCLUSION

We have developed an LLM-based framework that can translate a user's navigation instructions into paths that enable the robot to carry out the instructions precisely. To enhance the reliability of the LLM in translation, we have proposed methods to detect and address ambiguity and unknown user preferences in the instructions. `text-embedding-ada-002` combined with a random forest classifier was optimized for ambiguity detection, and

GPT-4 was utilized to solicit user inputs to clarify ambiguity and unknown preferences. To improve the user experience, we added a memory component to GPT-4 to retain information about the user's preferences for future interactions. By integrating with existing methods for translating the processed instructions into LTL specifications and computing robot navigation paths that meet these specifications, our framework offers an end-to-end capability to convert natural language instructions into viable navigation paths. Through performance comparisons with other models in ambiguity detection and validation in a robot navigation scenario, we have verified the effectiveness of our framework. Future research will explore integrating visual input to refine the robot's understanding of complex instructions and environments. Moreover, another direction will be integrating real-time feedback for dynamic path adjustment and expanding language support for broader applicability.

Fig. 3: Dialog between a user and a robot in the disambiguation and user preference identification process.

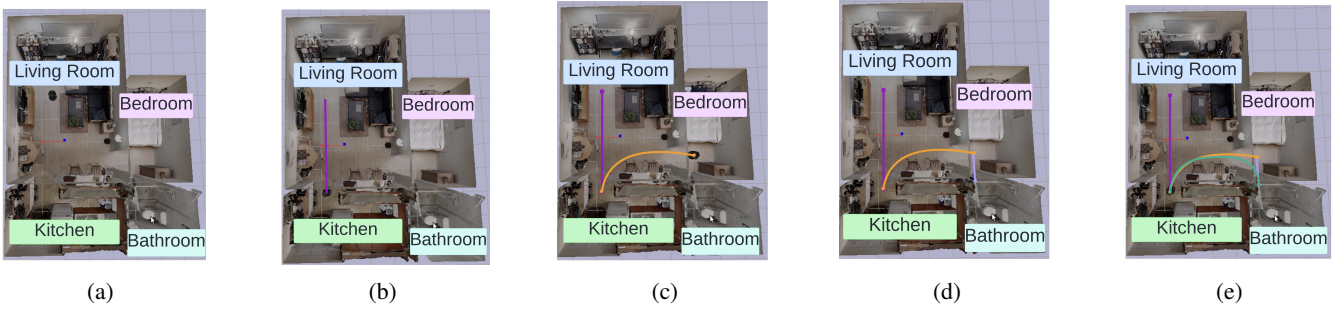| (a) | (b) | (c) | (d) | (e) |

Fig. 4: Validation of the proposed framework for robot navigation: Each figure illustrates a distinct stage in the robot's navigation as it follows the user's instructions (refer to Fig. 3 for the instructions.)

REFERENCES

[1] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[2] OpenAI, "Gpt-4 technical report," https://arxiv.org/abs/2303.08774, 2023.

[3] T. B. Brown *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[4] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 730–27 744.

[5] A. Chowdhery *et al.*, "Palm: scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, no. 1, mar 2024.

[6] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *ArXiv*, vol. abs/2307.09288, 2023.

[7] R. Bommasani *et al.*, "On the opportunities and risks of foundation models," *ArXiv*, 2021.

[8] A. Mallen *et al.*, "When not to trust language models: Investigating effectiveness of parametric and non-parametric memories," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 9802–9822.

[9] K. Tian *et al.*, "Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5433–5442.

[10] M. Zhang *et al.*, "How language model hallucinations can snowball," 2023.

[11] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," *Robotics: Science and Systems*, 2021.

[12] F. Duvallet *et al.*, "Inferring maps and behaviors from natural language instructions," in *Experimental Robotics - The 14th International Symposium on Experimental Robotics, ISER 2014, June 15-18, 2014, Marrakech and Essaouira, Morocco*, ser. Springer Tracts in Advanced Robotics, M. A. Hsieh, O. Khatib, and V. Kumar, Eds., vol. 109. Springer, 2014, pp. 373–388.

[13] S. Huang *et al.*, "Instruct2Act: Mapping Multi-modality Instructions to Robotic Actions with Large Language Model," May 2023, arXiv:2305.11176 [cs].

[14] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2Motion: from natural language instructions to feasible plans," *Autonomous Robots*, vol. 47, no. 8, pp. 1345–1365, Dec. 2023.

[15] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 23 171–23 181.

[16] M. Ahn *et al.*, "Do As I Can, Not As I Say: Grounding Language in Robotic Affordances," Aug. 2022, arXiv:2204.01691 [cs].

[17] W. Huang *et al.*, "Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents," Mar. 2022, arXiv:2201.07207 [cs].

[18] J. Park *et al.*, "Clara: Classifying and disambiguating user commands for reliable interactive robotic agents," *IEEE Robotics and Automation Letters*, 2023.

[19] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[20] F. I. Doğan *et al.*, "Asking Follow-Up Clarifications to Resolve Ambiguities in Human-Robot Conversation," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2022, pp. 461–469.

[21] F. Doshi *et al.*, "Spoken language interaction with model uncertainty: an adaptive human–robot interaction system," *Connection Science*, vol. 20, no. 4, pp. 299–318, 2008.

[22] J. Wu *et al.*, "Tidybot: Personalized robot assistance with large language models," *Autonomous Robots*, 2023.

[23] Y. Chen, R. Gandhi, Y. Zhang, and C. Fan, "NL2TL: Transforming natural languages to temporal logics using large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 15 880–15 903.

[24] J. Pan *et al.*, "Data-Efficient Learning of Natural Language to Linear Temporal Logic Translators for Robot Task Specification," Mar. 2023, arXiv:2303.08006 [cs].

[25] A. Mavrogiannis *et al.*, "Cook2LTL: Translating Cooking Recipes to LTL Formulae using Large Language Models," Sep. 2023, arXiv:2310.00163 [cs].

[26] V. Kurtz and H. Lin, "Temporal logic motion planning with convex optimization via graphs of convex sets," *IEEE Transactions on Robotics*, vol. 39, pp. 3791–3804, 2023.

[27] OpenAI, "New embedding models and api updates," https://openai.com/blog/new-embedding-models-and-api-updates, Jan. 2024, accessed: 2024-03-15.

[28] M. U. Aytekin and O. Ayhan Erdem, "Generative pre-trained transformer (gpt) models for irony detection and classification," in *2023 4th International Informatics and Software Engineering Conference (IISEC)*, 2023, pp. 1–8.

[29] N. Gopalan, D. Arumugam, L. Wong, S. Tellex *et al.*, "Sequence-to-sequence language grounding of non-markovian task specifications," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[30] Y. Oh, R. Patel, T. Nguyen, B. Huang, E. Pavlick, and S. Tellex, "Planning with State Abstractions for Non-Markovian Task Specifications," in *Proceedings of Robotics: Science and Systems*, Freiburg, Germany, June 2019.