The Conversation is the Command: Interacting with Real-World Autonomous Robot Through Natural Language

Linus Nwankwo
linus.nwankwo@unileoben.ac.at
Chair of Cyber-Physical Systems, Montanuniversität
Leoben, Austria

Elmar Rueckert
Chair of Cyber-Physical Systems, Montanuniversität
Leoben, Austria

ABSTRACT

In recent years, autonomous agents have surged in real-world environments such as our homes, offices, and public spaces. However, natural human-robot interaction remains a key challenge. In this paper, we introduce an approach that synergistically exploits the capabilities of large language models (LLMs) and multimodal vision-language models (VLMs) to enable humans to interact naturally with autonomous robots through conversational dialogue. We leveraged the LLMs to decode the high-level natural language instructions from humans and abstract them into precise robot actionable commands or queries. Further, we utilised the VLMs to provide a visual and semantic understanding of the robot's task environment. Our results with 99.13% command recognition accuracy and 97.96% commands execution success show that our approach can enhance human-robot interaction in real-world applications. The video demonstrations of this paper can be found at https://osf.io/wzyf6 and the code is available at our repository[1].

CCS CONCEPTS

• **Human-centered computing** → **Human computer interaction (HCI)**; **Interaction paradigms**; **Natural language interfaces**;

KEYWORDS

Human-robot interaction, LLMs, VLMs, ChatGPT, ROS, autonomous robots, natural language interaction.

ACM Reference Format:
Linus Nwankwo and Elmar Rueckert. 2024. The Conversation is the Command: Interacting with Real-World Autonomous Robot Through Natural Language. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion), March 11–14, 2024, Boulder, CO, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3610978.3640723

1 INTRODUCTION

The exploration of human-robot interaction (HRI) [29], [32] and its advancement into real-world applications has been a topic of significant research over the past decades [30]. Current approaches for controlling and interacting with autonomous robots in the real world have been dominated by complex teleoperation controllers [13], teach pendants [2], and rigid command protocols [16], where the robots execute predefined tasks based on specialized programming languages. As the challenges we present to these robots become more intricate and the environments they operate in grow more unpredictable [18], there arises an unmistakable need for more natural and intuitive interaction mechanisms.

Prior works have seen a tilt towards techniques like reinforcement [21] and imitation learning [2]. By leveraging iterative learning and human demonstrations, these strategies have shown a capacity for fostering nuanced robot behaviours, as demonstrated in [28]. However, the often computational burdens [15], and the high costs associated with reward specification [25], task-specific training, or fine-tuning, commonly observed in the reinforcement and imitation learning frameworks, have limited the practical applicability of these methods, especially for simpler robotic tasks.

Prompted by these challenges, we turned our focus to the recent advancement in large language models (LLMs) [23], [6] and multi-modal vision-language models (VLMs) [22], [24] to foster an intuitive human-robot collaboration. This paper introduces an innovative approach that exploits the inherent natural language capabilities of pre-trained LLMs and VLMs to enable humans to interact with autonomous robots through natural language dialogues. As demonstrated in Figure 1, we aim to realize a new approach to human-robot interactions—one where the conversation is the command (refer to Sections 3 & 4 for more details).

Our contributions are therefore threefold: (a) we introduced a framework that can leverage independent pre-trained LLMs (e.g., OpenAI GPT-2 [23] & GPT-3 [6], Google BERT [8], Meta AI LLaMA [31], etc), and VLMs (e.g., CLIP[22]) to enable real-world autonomous robots to interact with humans or other entities using natural language dialogues. (b) we performed real-world experiments with our developed framework to ensure that the robot's actions are always aligned with the user's instructions, thereby reducing the likelihood of erroneous operations. (c) we have made our code and associated resources available to the public. This allows for easy reproducibility of our results.

2 RELATED WORK

The recent rise of natural language processing (NLP) [33], marked by large language models (LLMs) like OpenAI GPT-3 [6], Google BERT [8], HuggingFace distilBERT [26], EleutherAI GPTNeoX [3], Meta AI LLaMA [31], Facebook RoBERTa [14], multi-modal vision-language models (VLMs) e.g., CLIP [22], DALL-E [24], and their successors, has opened new avenues for human-robot interaction. The inherent capacity of these models to understand and generate

[1] https://github.com/LinusNEP/TCC_IRoNL.git

HRI '24 Companion, March 11–14, 2024, Boulder, CO, USA

ACM ISBN 979-8-4007-0323-2/24/03.
https://doi.org/10.1145/3610978.3640723

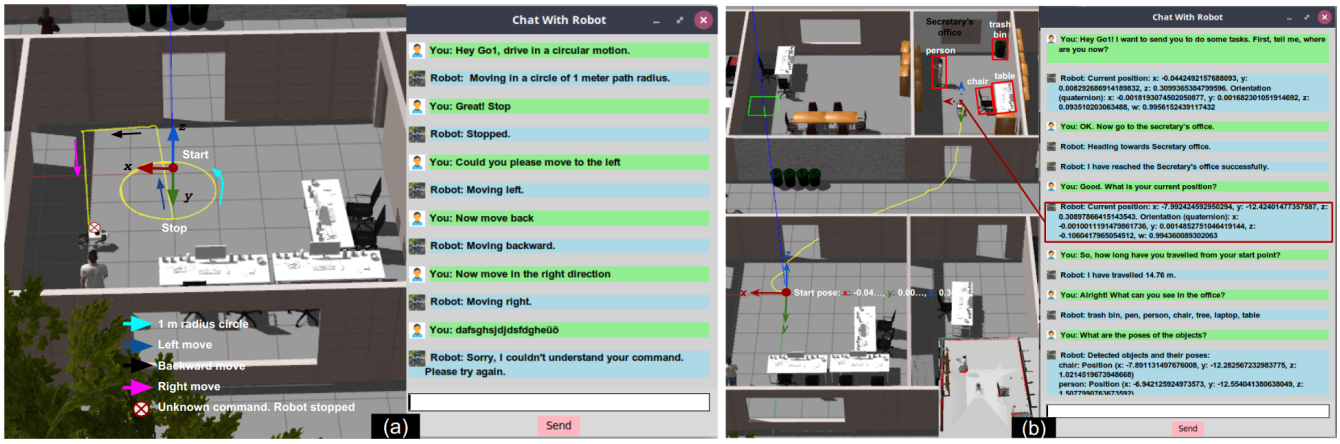

Figure 1: Example demonstration of our framework. We demonstrated these results in the real world as shown in the summary video at https://osf.io/wzyf6. In (a), our framework decodes the high-level instructions such as *"move in a circular pattern", "move forward, go right, etc."* from humans, and abstracts them to the robot's physical actions. In (b), we leveraged our framework for the robot's task environment understanding, information requests, and goal navigation.

human-like text as well as visual observations has led to several interesting applications [4], [5]. Recent works such as [1], [5], [10] and [12] have successfully incorporated LLMs and VLMs into robotic systems, allowing the LLMs to interpret and execute complex commands. Similarly, Wangchunshu et al.[35], Kaiwen Zhou et al. [34], and Miguel et al. [9] in their work demonstrated how LLMs could be used to facilitate real-time feedback, zero-shot object navigation and cognitive learning in autonomous robots.

While these works are exceptional, they have focused solely on a step-by-step task description [12], and rely completely on the LLM's ability to plan the robot's actions and act. In most complex real-world scenarios, especially as LLMs can sometimes hallucinate [7] or generate inconsistent data, their approach may introduce inconsistencies and randomness in the robot's actions.

On the contrary, our approach draws inspiration from the work of Yagi Xie et al. [33]. Instead of relying completely on the LLMs' ability to plan and execute the robot's actions, we employ a bidirectional approach, simply using the LLMs as a linguistic decoder [33], and a classical robot operating system (ROS) [20] navigation planner to plan the robot's actions. We provide the LLMs with a dictionary of task descriptions and action patterns. We then use the ROS planner[2] to plan the actual physical actions of the robot (e.g., path planning, localisation, obstacle avoidance, mapping, etc.), as shown in Figure 1b.

3 METHODS

An overview of our framework's architecture is shown in Figure 2. One of our core objectives is to develop a framework that enables real-world autonomous robots to interact with humans or other entities using natural language dialogues. To achieve this objective, we decompose the task into three subtasks: (a) the integration of LLMs and VLMs, (b) the development of the robot execution mechanism (REM) node, and (c) the development of the chat graphical user interface (ChatGUI). An overview is shown in Figure 2. This section provides details of each of the above three subtasks.

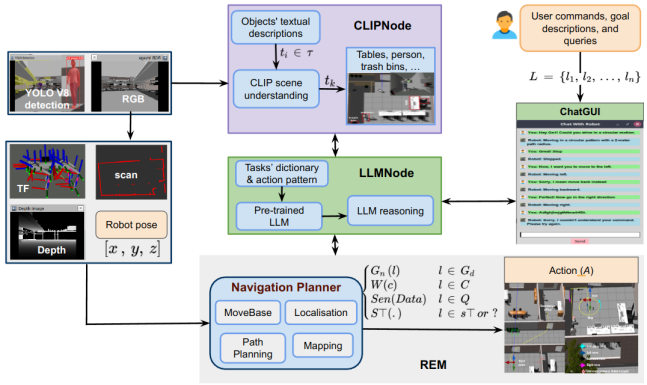

Figure 2: Overview of our framework's architecture. The LLMNode decodes the natural language conversations. The CLIPNode provides a visual and semantic understanding of the robot's task environment. The REM node abstracts the high-level understanding from the LLMNode to actual robot actions. The ChatGUI serves as the user's primary interaction point. See Subsections 3.1, 3.2, and 3.3 for more details.

3.1 Integration of LLMs and VLMs

To decode the natural language conversation and abstract them to the robot's actions, we developed a ROS node, LLMNode (light green block in Figure 2) to establish communication between the pre-trained LLMs and the rest interfaces within ROS ecosystem [20]. The LLMNode subscribes to topics that provide essential data, e.g., odometry for spatial sensing and outputs from the CLIPNode (light purple block in Figure 2) for visual observation and object recognition. We used the LLMNode to handle incoming natural language inputs from the ChatGUI (Subsection 3.3) by first passing

[2] https://github.com/ros-planning/navigation

them through the pre-trained LLM [23]. The output is then mapped to the robot's actionable commands or queries.

In the mapping process, we leverage pattern matching to align the generated text with predefined actions or information requests. For example, navigation commands are translated into goals for the robot to pursue within its environment, while queries Q about the robot's status or surroundings are addressed with information derived from the robot's sensor data. The LLMNode also oversees the execution and feedback process. We added this function to provide real-time feedback through the publishing of messages, which not only inform the user but also log the interaction data for subsequent analysis (see Subsection 4.1 for more details).

Summarily, the LLMNode function can be described as a mapping from natural language inputs to robot actions, i.e., LLMNode : $\mathcal{L} \mapsto \mathcal{A}$ where $\mathcal{L}$ represents the space of natural language inputs and $\mathcal{A}$ denotes the set of possible robot actions. This mapping is a composition of several functions, as depicted in Eq. 1.

$$\text{LLMNode}(l) = \text{REM}(\text{LM}(l),\ \text{Sen}(\text{Data})),\ l \in \mathcal{L} \tag{1}$$

From Eq. 1, LM(l) is the language model's interpretation of the input $l_i \in \mathcal{L}$ and Sen(Data) represents the sensor data that informs the context of the command. The REM node (Section 3.2) then translates this into an executable command for the robot.

Furthermore, to provide a visual and semantic understanding of the task environment (e.g., Figure 1b), we used the OpenAI contrastive language image pretraining (CLIP) model [22]. CLIP model consists of language and image encoders trained on a staggering 400 million image-text pairs [27]. Thus, we used it to encode the stream of RGB images from our observation source (Intel Realsense D435i) alongside the textual descriptions of objects in the image.

Formally, given an image $\mathcal{I}_t$ at time t, we consider a set of predefined textual descriptions $\mathcal{D} = \{d_1, d_2, ..., d_n\}$. Each description $d_i \in \mathcal{D}$ is mapped to a tokenized representation, forming a set $\mathcal{T} = \{t_1, t_2, ..., t_n\}$. This set encompasses human-readable labels for common office objects such as "table", "chair", "person", and so on. Using the CLIP model [22], we extract the feature vector of the image, i.e., $f_{\mathcal{I}} = \text{CLIP\_encode\_image}(\mathcal{I}_t)$. For each tokenized description $t_i \in \mathcal{T}$, its feature vector is obtained as $f_{\mathcal{T}_i} = \text{CLIP\_encode\_text}(t_i)$. Subsequently, for the image feature and every text feature, we compute the similarity scores using $\mathcal{S}_i = f_{\mathcal{I}} \cdot f_{\mathcal{T}_i}^{\mathcal{T}}$. Thus, we determine the recognized object within the image by selecting the textual description that yields the highest similarity score, as depicted in Eq. 2.

$$\text{Recognized Object} = t_k;\ k = \arg\max_{i=1}^{n} \mathcal{S}_i \tag{2}$$

Additionally, our model uses the bounding boxes from YOLO V8 [11] to determine regions of interest (ROI) within the image. Notably, the centres of these bounding boxes are employed as the spatial coordinates for the recognized objects, capturing both the identity and the location of the objects in the scene.

We embodied the entire process within a ROS node, CLIPNode (light purple block in Figure 2). The output from Eq. 2, representing the recognized objects along with their respective spatial coordinates, is published as a ROS [20] topic. These are subsequently subscribed to by the LLMNode to handle the natural language commands, generate responses, and decide on actions for the robot to take. For instance, based on the prompt used by the robot's user, it can direct the robot to navigate to a detected object or provide information about detected objects and their positions.

3.2 Robot Execution Mechanism (REM)

To abstract the high-level language understanding and environment sensing from the LLMNode to actual robot actions, we developed the robot execution mechanism (REM) node. This node translates intents extracted from the LLMNode into actionable tasks for physical execution by the robot. Central to the REM node's functionality is processing navigation goals, $\mathcal{G}_n$ (e.g., Figure 1b). When a textual description of a goal destination $\mathcal{G}_d$, such as "navigate to the Secretary's office" is provided, the REM node translates this into precise goal coordinates (x_l, y_l, z_l, w_l) within the robot's operational environment via a mapping process that correlates the descriptive labels with their corresponding spatial coordinates i.e., $\mathcal{G}_d \mapsto (x_l, y_l, z_l, w_l)$. To navigate the robot to the goal, we used the *MoveBase* package of the ROS navigation planner, which provides an action server for handling navigation goals. REM node sends the goal to this server and monitors its progress.

Besides navigation goals, the REM node also handles custom movement commands C (e.g., Figure 1a) which are not tied to specific goal locations, but rather to particular motion patterns, such as "rotate in place", "move forward" etc. We encoded these patterns into the robot's YAML configuration files, allowing for a flexible command set $c_i \in C$ that can be expanded or modified as required. We use the REM node to translate the commands into **Twist** messages $\mathcal{W}$ with linear (v_x, v_y, v_z) and angular $(\omega_x, \omega_y, \omega_z)$ velocity components as $\mathcal{W}(c) = Twist(v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)$.

In addition to handling movement, we integrated a security measure to halt the robot when an unrecognized command (e.g., the last command in Figure 1a) is received, issuing zero velocities to stop all motion, ensuring safe operation.

$$\text{REM}_l = \begin{cases} \mathcal{G}_n(l), & \text{if } l \in \mathcal{G}_d \\ \mathcal{W}(c), & \text{if } l \in C \\ \text{Sen(Data)}, & \text{if } l \in Q \\ Stop(), & \text{if } l \in \text{stop or unknown command} \end{cases} \tag{3}$$

Formally, as summarised in Eq. 3, the REM node abstracts the complexity of the robot navigation and command execution, translating the high-level instructions into physical actions.

3.3 Chat Interface Development

To provide an intuitive conversational platform that would facilitate natural language interaction between the robot and its human users, we developed a simple and user-friendly chat graphical user interface (ChatGUI) which serves as the user's primary interaction point with the robot through textual communication. We designed the ChatGUI using Tkinter libraries[3] and integrated it within ROS [20] for message passing. We employed the standard ROS publish/subscribe communication mechanisms for the ChatGUI development, specifically, a bidirectional message exchange approach, i.e., ChatGUI : UserInputs ↔ LLMNodeOutputs. User natural language inputs are published to the LLMNode, and the responses are subscribed to and displayed to the user on the ChatGUI interface.

[3] https://docs.python.org/3/library/tkinter.html

We developed the ChatGUI with event-driven architecture to ensure that user actions, such as sending a message or issuing a command, trigger corresponding updates in the ChatGUI or result in the publishing of commands to the LLMNode. We encapsulated this process in a function that translates user actions into corresponding LLMNode responses.

4 PRELIMINARY RESULTS

We conducted real-world and simulated experiments to demonstrate the applicability and adaptability of our framework. For simulation, we used the Unitree Go1 Gazebo packages[4] and a ROS-based open-source mobile robot adapted from [17]. We ran all the simulations with a ground station PC with Nvidia Geforce RTX 3060 Ti GPU, 8GB memory running Ubuntu 20.04, ROS Noetic distribution.

In real-world experiments, we used a Lenovo ThinkBook Intel Core i7 with Intel iRIS Graphics running Ubuntu 20.04, ROS Noetic distribution. Unitree Go1 quadruped robot was used. The robot is equipped with Intel Realsense D435i RGB-D camera and Ouster 3D LiDAR for both visual and spatial observations of the task environment. All the real-world experiments were performed in our laboratory office (11 rooms) and outside corridor environment, measuring approximately 18×20 m and 6×120 m respectively.

We experimented with different pre-trained open-source LLMs like OpenAI GPT-2 [23], Google BERT [8], HuggingFace distilBERT [26], EleutherAI GPTNeoX [3], Meta AI LLaMA [31], and Facebook RoBERTa [14]. OpenAI GPT-3 [6] and GPT-4 [19] are also adaptable to our framework. However, due to their API access limitations, we mostly utilised the open-access and free versions of the LLMs (GPT-2 [23] specifically) in our experiments.

4.1 Initial Evaluation / Participants

In our initial evaluation, we invited 21 participants (mostly students) with an average age of 23 (±5) and gender distribution, 61.9% male, 28.6% female and 9.5% others to assess the intuitiveness of our framework by interacting with the robots using natural language. We instructed the participants to command the robots to navigate to locations, identify objects, and inquire about their status. We meticulously logged the interaction data which includes the participant's input text, the LLM's output, the true label, the LLMNode predicted label, etc. To quantitatively evaluate the performance of our framework, we established four key metrics:

(a) Command Recognition Accuracy (CRA): With the CRA, we assess how accurately the LLMNode interprets the natural language commands. This aids us in pinpointing instances where the predicted label diverged from the true label, providing insights into potential areas for improvement. **(b) Object Identification Accuracy (OIA)**: We employed this metric to assess the precision of the CLIPNode in identifying and localizing objects within the robot's task environment. **(c) Navigation Success Rate (NSR)**: We utilised this metric to determine the effectiveness of the REM node in successfully navigating the robot to the designated locations. **(d) Average Response Time:(ART)**: We logged in ROS Unix epoch clock standard, the time a message is sent from the ChatGUI, the time it is received by the LLMNode, and the time the robot responds. With the ART, we compute the average duration from receiving the user's chat command to initiating the robot's movement.

Figure 3 presents our preliminary statistical results obtained from the interaction data analysis. The top row of Figure 3 shows the performance metrics and the confusion matrix (for selected labels) of the LLMNode. The CRA, with a prediction accuracy of 99.13% (i.e., how often the "Predicted labels" matched the "True labels"), indicates a high level of accuracy in the command interpretation. This reflects the robustness of the LLMNode in processing the natural language inputs. The OIA on the other hand, achieved 55.20% accuracy, indicating room for improvement in our CLIPNode integration. Further, the NSR at 97.96%, indicates good performance in the REM's ability to abstract the high-level understanding from the LLMNode to the actual robot's navigation actions. Also, the

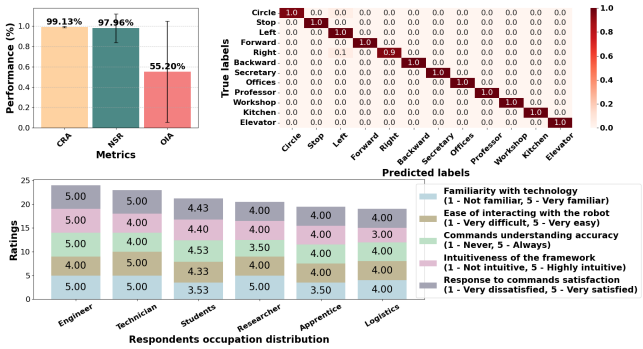

Figure 3: Performance and variability measures illustrating CRA, OIA, and NSR (top) and the participants' feedback (bottom) based on the logged interaction data.

overall ART across all the selected commands (refer to the figure at https://osf.io/ufctx) is approximately 0.45 seconds. This indicates that, on average, the robot takes less than half a second from receiving a chat command to initiating movement, which suggests a relatively quick response time for our framework.

Furthermore, the bottom row of Figure 3 shows the participants' feedback (refer to the questionnaire at https://osf.io/dgbtr). With 4 and 5 ratings as favourable benchmarks, 80.9% and 76.2% of the participants respectively rated the ease of interaction and the intuitiveness of our framework as favourable, while 85.7% are satisfied with the response of the robot to their natural language commands.

5 CONCLUSION AND FUTURE WORK

We introduced a framework that leverages the inherent capabilities of large language models (LLMs) and multimodal vision-language models (VLMs) to enhance human-robot interaction through natural conversation. Our evaluation from the logged interaction data and participants' feedback was overwhelmingly positive. The high command recognition accuracy and effective task execution, show that our framework can simplify human-robot interaction. Looking ahead, we aim to refine the framework across several dimensions, not just for ROS-based autonomous robots. The CLIPNode will be further improved for broader object recognition, and the LLMNode will be fine-tuned with domain-specific data for better contextual and voice understanding. User experience will be a priority, with a focus on creating a more intuitive and adaptive chat interface.

[4] https://github.com/unitreerobotics/unitree_guide

REFERENCES

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. arXiv:2204.01691 [cs.RO]

[2] Baris Akgun, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz. 2012. Trajectories and Keyframes for Kinesthetic Teaching: A Human-Robot Interaction Perspective. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction* (Boston, Massachusetts, USA) *(HRI '12)*. Association for Computing Machinery, New York, NY, USA, 391–398. https://doi.org/10.1145/2157689.2157815

[3] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. arXiv:2204.06745 [cs.CL]

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. arXiv:2307.15818 [cs.RO]

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023. RT-1: Robotics Transformer for Real-World Control at Scale. arXiv:2212.06817 [cs.RO]

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL]

[7] Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination Detection: Robustly Discerning Reliable Answers in Large Language Models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 245–255. https://doi.org/10.1145/3583780.3614905

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL]

[9] Miguel Á. González-Santamarta, Francisco J. Rodríguez-Lera, Ángel Manuel Guerrero-Higueras, and Vicente Matellán-Olivera. 2023. Integration of Large Language Models within Cognitive Architectures for Autonomous Robots. arXiv:2309.14945 [cs.RO]

[10] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents. arXiv:2201.07207 [cs.LG]

[11] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics

[12] Teyun Kwon, Norman Di Palo, and Edward Johns. 2023. Language Models as Zero-Shot Trajectory Generators. arXiv:2310.11604 [cs.RO]

[13] Tsung-Chi Lin, Achyuthan Unni Krishnan, and Zhi Li. 2023. Perception-Motion Coupling in Active Telepresence: Human Behavior and Teleoperation Interface Design. *J. Hum.-Robot Interact.* 12, 3, Article 31 (mar 2023), 24 pages. https://doi.org/10.1145/3571599

[14] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL]

[15] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. 2022. Interactive Language: Talking to Robots in Real Time. arXiv:2210.06407 [cs.RO]

[16] Debasmita Mukherjee, Kashish Gupta, and Homayoun Najjaran. 2022. A Critical Analysis of Industrial Human-Robot Communication and Its Quest for Naturalness Through the Lens of Complexity Theory. *Frontiers in Robotics and AI* 9 (2022). https://doi.org/10.3389/frobt.2022.870477

[17] Linus Nwankwo, Clemens Fritze, Konrad Bartsch, and Elmar Rueckert. 2023. ROMR: A ROS-based open-source mobile robot. *HardwareX* 14 (2023), e00426. https://doi.org/10.1016/j.ohx.2023.e00426

[18] Linus Nwankwo and Elmar Rueckert. 2023. Understanding Why SLAM Algorithms Fail in Modern Indoor Environments. In *Advances in Service and Industrial Robotics*, Tadej Petrič, Aleš Ude, and Leon Žlajpah (Eds.). Springer Nature Switzerland, Cham, 186–194.

[19] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[20] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Ng. 2009. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software* 3.

[21] Ahmed Hussain Qureshi, Yutaka Nakamura, Yuichiro Yoshikawa, and Hiroshi Ishiguro. 2018. Intrinsically motivated reinforcement learning for human–robot interaction in the real-world. *Neural Networks* 107 (2018), 23–33. https://doi.org/10.1016/j.neunet.2018.03.014 Special issue on deep reinforcement learning.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV]

[23] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. https://api.semanticscholar.org/CorpusID:160025533

[24] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-Shot Text-to-Image Generation. arXiv:2102.12092 [cs.CV]

[25] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. 2023. Vision-Language Models are Zero-Shot Reward Models for Reinforcement Learning. arXiv:2310.12921 [cs.LG]

[26] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL]

[27] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. arXiv:2111.02114 [cs.CV]

[28] Saurav Singh and Jamison Heard. 2022. Human-Aware Reinforcement Learning for Adaptive Human Robot Teaming. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction* (Sapporo, Hokkaido, Japan) *(HRI '22)*. IEEE Press, 1049–1052.

[29] Hang Su, Wen Qi, Jiahao Chen, Chenguang Yang, Juan Sandoval, and Med Amine Laribi. 2023. Recent advancements in multimodal human–robot interaction. *Frontiers in Neurorobotics* 17 (2023). https://doi.org/10.3389/fnbot.2023.1084000

[30] Andrea Thomaz. 2023. Robots in Real Life: Putting HRI to Work. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction* (Stockholm, Sweden) *(HRI '23)*. Association for Computing Machinery, New York, NY, USA, 3. https://doi.org/10.1145/3568162.3578810

[31] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL]

[32] Katie Winkle, Erik Lagerstedt, Ilaria Torre, and Anna Offenwanger. 2023. 15 Years of (Who)Man Robot Interaction: Reviewing the H in Human-Robot Interaction. *J. Hum.-Robot Interact.* 12, 3, Article 28 (apr 2023), 28 pages. https://doi.org/10.1145/3571718

[33] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. 2023. Translating Natural Language to Planning Goals with Large-Language Models. arXiv:2302.05128 [cs.CL]

[34] Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. 2023. ESC: Exploration with Soft Commonsense Constraints for Zero-Shot Object Navigation. In *Proceedings of the 40th International Conference on Machine Learning* (Honolulu, Hawaii, USA) *(ICML'23)*. JMLR.org, Article 1806, 14 pages.

[35] Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, Shiding Zhu, Jiyu Chen, Wentao Zhang, Ningyu Zhang, Huajun Chen, Peng Cui, and Mrinmaya Sachan. 2023. Agents: An Open-source Framework for Autonomous Language Agents. arXiv:2309.07870 [cs.CL]