ABSTRACT
With the increasing prevalence and diversity of robots interacting in the real world, there is need for flexible, on-the-fly planning and cooperation. Large Language Models are starting to be explored in a multimodal setup for communication, coordination, and planning in robotics. Existing approaches generally use a single agent building a plan, or have multiple homogeneous agents coordinating for a simple task. We present a decentralised, dialogical approach in which a team of agents with different abilities plans solutions through peer-to-peer and human-robot discussion. We suggest that argument-style dialogues are an effective way to facilitate adaptive use of each agent’s abilities within a cooperative team. Two robots discuss how to solve a cleaning problem set by a human, define roles, and agree on paths they each take. Each step can be interrupted by a human advisor and agents check their plans with the human. Agents then execute this plan in the real world, collecting rubbish from people in each room. Our implementation uses text at every step, maintaining transparency and effective human-multi-robot interaction.

KEYWORDS
Mixed Human-Robot teams; Multi-robot coordination and collaboration; Large Language Models

1 INTRODUCTION
Multirobot systems, while still reasonably rare in everyday life, are becoming increasingly common in domains such as agriculture [1, 11], search and rescue [5], and construction [20]. It is projected that the household robotics market will grow considerably this decade [18], leading to large numbers of robots interacting not only with their users but also with each other. This presents increasing demand for robotic systems which can flexibly adapt to new tasks without prior training. There is also a recent trend towards generalist robotics; the same model or overarching structure can be deployed on a wide variety of platforms and tasks [14]. To this end, it is desirable to move towards a domain-agnostic coordination structure that is common across platforms so that the group can be treated as a single entity. A popular area of recent development in the AI landscape is that of Large Language Models (LLMs) [21]; agents that operate in the domain of text through next-symbol prediction. LLMs may present an effective way to interpret task or agent descriptions, as well as calling on internalised understanding and reasoning to develop the available information. LLMs can also facilitate some degree of communication between agents that helps them organise while keeping the control flow understandable and accessible to a human operator. Multi-modal AI development is now influencing robotics and other agents-based research fields [6] by using LLMs to power end-to-end approaches which can understand and use human-written inputs to inform their actions [2]. This fits into an overarching vision towards a multimodal, generalist agent which can understand and operate on text, images, and other inputs in robotics [16]. Although conversational agents such as ChatGPT are typically used to model a human-agent conversation, some works have focused on modelling a conversation between multiple agents. This is typically done through “role-playing”; an agent is told to “imagine” that it is a person with a certain role and then enters a conversation whilst assuming that role [10]. This process can be used to model internal monologue for a single agent who “talks to themselves” about what they can perceive and do [9]. Conversational approaches can also use multiple personas, each of whom brings a different specialised perspective to the collective generation of text by editing the group solution to fulfill different goals that they each have for the end result [19].

Role-playing has been used for a variety of tasks including debate [3], auctions, haggling [7, 13], and checking with a human supervisor that they are not hallucinating [17]. These approaches set the conditions for a dialogue and leave the agents to talk, assuming that an intelligent solution emerges naturally from the conversation. This has been used to create a team of software developers who each write code and pass responsibilities to the next developer [8]. LLMs have also been integrated into robotic simulators for communication to organise who performs each task, allowing robots to coordinate their workspaces and decide which of them is able to reach an object [12]. A similar approach simulates agents fetching items in a house, they communicate and incorporate short-term memory to request assistance from each other on the fly [22]. Some works include diverse skillsets where each agent has different capabilities and skills, such as a work where the agents build a team with the required skills before planning and acting [4].

This work is licensed under a Creative Commons Attribution International 4.0 License.

We present a proof-of-concept system that leverages the knowledge of pretrained models by building language-based agents which talk to each other, and with humans, using natural language. This allows agents to discuss and debate their strategies towards a collaborative solution to a high-level mission objective with observation or assistance from a human supervisor. This forms a pipeline that allows agents to take a high-level task description and autonomously perform every step of the process, from planning to assignment.

2 DEMONSTRATION

An LLM is used to create a conversation where agents build paths to be executed on hardware, which in turn can detect problems and prompt the LLM to re-plan. The pipeline uses text at every step of the process to retain deep meaning from end to end. A Python program takes human input, and calls the GPT API (“gpt-4-vision-preview”) [15] for language generation. The conversation produces plans which are passed directly to the robots. In brief, the steps are:

1. **Agent Ego**: The “system message” (identity) for each agent is set to a description of each agent, plus some general guidance on how to debate.
2. **Environment Description**: We provide a flowchart-style environment model that shows agents which rooms are connected with arrows (see Fig.2c).
3. **Human Supervisor**: The supervisor presents a task to the agents.
4. **Discussion**: Agents discuss the task and plan their approach.
5. **Calling the supervisor**: Agents call the supervisor when done, or if they need help. The supervisor can ask for alterations if desired or approve the plan. Agents are added to the chat and a human supervisor is given a chat box to speak with them. When agents discuss, they start each message with their name and it is added to the log. From each agent’s perspective, they perceive every other agent’s message as one from the human, but the name tags allow them to understand the conversation properly. The system prompt, which the LLM considers with every message, encourages agents to negotiate with each other. This is important because the default GPT configuration is polite and rarely contradicts its interlocutor, however for collaborative planning agents should point out mistakes. When they reach a decision, a path of rooms for each agent is extracted and passed to hardware for execution. Two TurtleBot3 robots (see Fig.2a) are equipped with a LIDAR and an optical camera. ROS gathers data from the LIDAR and controls the differential-drive motors. The LIDAR is used for collision avoidance to protect the hardware as it moves around the environment. The optical camera is used to detect ArUco markers which indicate rooms.

We demonstrate a conversational planning system deployed on real robots to simulate autonomous waste collection. A bin is mounted on top of each robot using a 3D-printed structure (see Fig.2b), turning the robots into mobile bins (see Fig.2a). Participants can engage by typing a task into the supervisor PC, and then watching the agents converse. The participants can also offer advice to the robots or point out issues before approving the plan. The robots then move around the arena to solve the task in real time. Participants are encouraged to provide vague and unusual instructions or interrupt the execution with novel information to test the system and explore the challenges of the approach as well as its potential. A cut-down example of this process is shown in Fig.1, where the agents must clean up after lunch at a conference. They define two roles, correct a mistake, and decide on the division of labour.

3 CONCLUSION

Robotic systems would benefit greatly from being able to understand, interpret, and utilise otherwise ignored text data. Inter-agent conversation may be a useful tool for decentralised mission planning with a human in the loop. We demonstrate conversational multi-agent coordination that allows agents to be represented with few-word names and calls on the deep knowledge of Large Language Models. The language-based approach can leverage expert opinion across many scenarios as the entire system is understandable and can be interfaced directly by a human user as much as is required. The proposed demonstration uses a language model to allow two robots to plan and execute a garbage collection task with a human supervisor in the loop and a large screen will display the conversations. Participants can interact and interrupt the system with different textual inputs to learn more about the capabilities and limitations of using language models for multi-robot coordination.

ACKNOWLEDGMENTS

This project was done as part of the Fast-Pi project funded by the UKRI Trustworthy Autonomous Systems Hub [EP/V00784X/1]. It was also supported by UK Research and Innovation [EP/502429/1].

1 Our demo video can be found here: https://www.youtube.com/watch?v=VCwG8aLvw
REFERENCES


