Abstract Meaning Representation for Grounded Human-Robot Communication

Claire Bonial[1], **Julie Foresta**[1], **Nicholas C. Fung**[1], **Cory J. Hayes**[1],
Philip Osteen[1], **Jacob Arkin**[2], **Benned Hedegaard**[2], **Thomas M. Howard**[2]
[1] Army Research Lab, [2] University of Rochester
claire.n.bonial.civ@army.mil,

Abstract

To collaborate effectively in physically situated tasks, robots must be able to ground concepts in natural language to the physical objects in the environment as well as their own capabilities. We describe the implementation and the demonstration of a system architecture that supports tasking robots using natural language. In this architecture, natural language instructions are first handled by a dialogue management component, which provides feedback to the user and passes executable instructions along to an Abstract Meaning Representation (AMR) parser. The parse distills the action primitives and parameters of the instructed behavior in the form of a directed a-cyclic graph, passed on to the grounding component. We find AMR to be an efficient formalism for grounding the nodes of the graph using a Distributed Correspondence Graph. Thus, in our approach, the concepts of language are grounded to entities in the robot's world model, which is populated by its sensors, thereby enabling grounded natural language communication. The demonstration of this system will allow users to issue navigation commands in natural language to direct a simulated ground robot (running the Robot Operating System) to various landmarks observed by the user within a simulated environment.

1 Introduction

Robots are increasingly used for their potential in disaster relief and search and rescue tasks (Murphy, 2014). There is a clear benefit to this, as robots can be used to provide aid and give situational awareness of the environment to people, who can remain at a safe distance and use information gathered by the robot to knowledgeably address the situation. Using robots in this way has required advances in robotics; however, robots in the current paradigm are still treated more as tools—often requiring human teleoperation, which inhibits the operator's awareness of their own immediate surroundings in potentially dangerous situations. The ability to speak to a robot as one would another human teammate would reduce the training time and cognitive burden on the operator, making the collaborative response more efficient. While there have also been relevant advances in task-oriented dialogue systems, such as Siri and Alexa, as well as widespread interest in systems leveraging large language models such as ChatGPT, these systems are limited in their applicability to physically situated tasks because they do not address grounding natural language to the physical environment of an embodied platform. In this paper, we describe a novel system architecture that supports grounded, bi-directional human-robot dialogue. This architecture is depicted in Figure 1.

In the sections to follow, we first provide a conceptual overview of the system capabilities (§2), and then detail the components of this architecture (§3) while highlighting the novel and primary contribution of the symbol grounding components: the Abstract Meaning Representation (AMR) (Banarescu et al., 2013) parser (§3.4), which we show to be uniquely suited to distill the action primitives and their parameters in a way that can be efficiently grounded, using our updated Distributed Correspondence Graph (DCG) (Howard et al., 2014) grounding component (§3.5). We then describe the demo (§4) and detail how distinct demo modes (§4.1) allow users to experience performance differences when the grounding component receives input from either a syntactic constituency parser or the meaning-based, AMR parser. We provide a brief comparison to related work (§5) and conclude with directions for ongoing and future work (§6).

Distribution Statement A. Approved for Public Release: Distribution Unlimited

Proceedings of the 4th International Workshop on Designing Meaning Representations, pages 34–44
June, 2023. ©2023 Association for Computational Linguistics

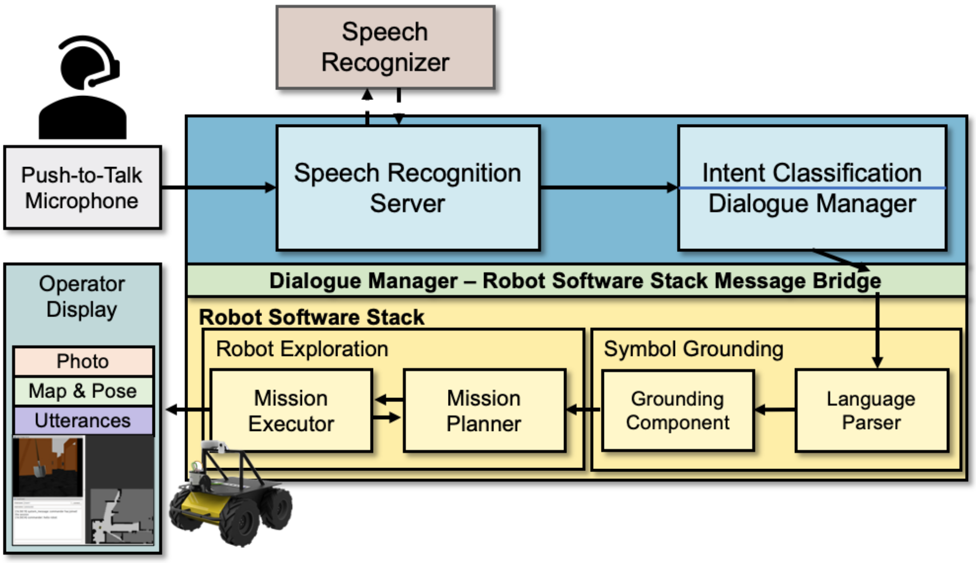

Figure 1: System architecture, supporting bi-directional, grounded communication between an operator and a remotely located robot.

2 System Capabilities

The implemented system in this research allows a human operator to speak to a remotely located robot in natural language, providing search and navigation instructions for the robot to execute. The current system has been successfully implemented for natural language control of a Clearpath Husky Unmanned Ground Vehicle (Clearpath Robotics, 2023) (shown in Figure 1), measuring about 39 inches in length and weighing about 110 pounds, which autonomously executes the natural language navigation instructions. In our implementation, the Husky is equipped with a LORD Microstrain 3DM-GDX5-25 IMU, Ouster OS1-64 Gen 1 Light Detection and Ranging (LIDAR) unit, and a Teledyne FLIR Blackfly GigE camera with a KOWA LMVZ41 high resolution camera lens. The robot computers consist of two Intel i7 equipped computers with NVIDIA 1650Ti graphics cards installed.

The robot runs on the Robot Operating System (ROS); thus, part of our research here includes creating a ROS wrapper around the AMR parsing component. The same ROS software stack can be used either within real-world robots or in simulation, and we have implemented and tested our architecture in both environments.

Because connectivity and bandwidth can be limited in disaster relief scenarios, our setup does not require internet connectivity, but it does currently require a stand-alone machine to run the natural language communication interface and dialogue management capabilities (shown in the top half of the architecture diagram in Figure 1), whereas the rest of the system architecture components run fully onboard the robot (shown in the bottom half of the architecture diagram in Figure 1).

3 System Components

In the following sections, we provide an overview of each of the architecture's components. We devote the most description to the primary novel contribution of this paper: the symbol grounding, which leverages an AMR parser together with a DCG grounding component.

3.1 Speech Recognition

The operator speaks to the robot using a microphone, currently implemented as the standard microphone capability of the computer running the user-facing dialogue interface components. The operator presses on an assigned key and speaks their instructions.

The speech recognition server listens to the user's speech and sends it to the speech recognizer component; we are currently leveraging the open-source Kaldi speech recognition toolkit (Povey et al., 2011). Kaldi provides automatic speech recognition (ASR), producing a text transcription of the user's speech. We selected Kaldi because we find that it gives relatively high-accuracy ASR but does not require internet connectivity.

3.2 Intent Classification & Dialogue Management

The text output from Kaldi is passed along to the joint intent classification and dialogue management component. This component has two elements: first, a classifier interprets the language with respect to the basic intent, and second, a dialogue manager dictates what the system should do next. For example, if the operator provides the instruction *Okay, Husky, check the path in front of you*, the system retrieves the most similar example to this seen in the training data, for example, *Scout the path in front*. The system would then provide an associated response message such as *executing* to provide feedback to the user. Finally, the system would pass the text instruction *Scout the path in front* along to the parsing component operating within the software stack for processing and eventual execution. This component is an adaptation of the Virtual Human Toolkit described in Hartholt et al. (2013), refined to support a robot platform (Marge et al., 2016).

Intent classification is treated as a retrieval problem, such that given the transcribed speech from the recognizer, the system can infer the intent by retrieving the most similar example from training data. The training data is organized into instruction-response pairs, where instructions are previously seen operator instructions, and responses are either messages sent back to the operator (such as feedback or clarification questions) or messages sent on to the robot software stack for further processing and execution. The training data instruction-response pairs are curated for a particular domain within a spreadsheet used to learn the weights of association such that a ranked list of potential matches is returned and the most similar instruction-response pair is selected (Leuski and Traum, 2011). In our implementation, the training data pairs are drawn from a corpus of human-robot collaborative dialogue for search and navigation, collected in a wizard-of-oz experimental paradigm (Marge et al., 2016) and subsequently annotated for relevant features of dialogue structure (Traum et al., 2018).

Dialogue management policies are defined based upon the matches obtained from the intent classifier, with two basic categories of response policies. The first is for actionable messages, where the robot is able to execute the instruction. For actionable commands, the basic policy is to jointly respond to the operator with feedback, demonstrating successful receipt of the instruction, and to send a simple text message of the instruction on to the robot software stack. The second policy is for non-actionable messages, which require clarification through further dialogue. The basic policy for non-actionable messages is to prompt the operator for clarification, such that any inability to infer the intent of the instruction can be overcome immediately through dialogue.

3.3 Message Bridge

A message bridge enabled by the Virtual Human Toolkit from Hartholt et al. (2013) connects the operator-facing natural language interface (which runs on a computer used by the operator) to the robot's software autonomy stack (which runs on the robot's onboard computer). The bridge enables connectivity between the two computers—sending synchronous messages from the operator-facing computer to the robot's computer and back again. Additionally, it enables the transfer between the two operating systems, where the output of the operator-facing computer is simply text, and is formatted as Robot Operating System (ROS) messages delivered to the software autonomy stack via a ROS topic for the robot to process.

3.4 Language Parser

We leverage the open-source AMR parser from Lindemann et al. (2019), specifically a model that has been retrained on a portion of the same human-robot dialogue corpus used to derive the instruction-response pairs described in §3.2. We selected this parser because the retrained model outperformed other competitive parsers retrained on the same small set of robot-directed instructions (Bonial et al., 2020), but we are working to make our implementation agnostic to any particular parser so that we can swap it out based on the current state of the art.

We implement wrapper code to interface the open-source AMR parser with ROS code that operates the automated systems aboard the robot including perception and motor control. The wrapper code takes in commands through ROS messages. These messages can be generated by the autonomy stack running on the robot, piped directly to the AMR parser as a string through ROS commands, or generated by other software. In our case, the dialogue manager generates these commands and the message bridge publishes them as a string to a

ROS topic. The command string is extracted from the ROS message and used as input for the AMR parser.

Thus, the parser accepts the text instructions output by the dialogue manager, and parses this into an AMR directed, a-cyclic graph (DAG). Because AMR abstracts away from some idiosyncratic linguistic variation in favor of representing core concepts, the AMR parse is a very effective distillation of action primitives and the parameters of that action. For example, regardless of whether the operator instructs the robot to *Drive to the barrel on the left* or *Take a drive to the left barrel*, these instructions will be encoded with identical AMR graphs, shown in Figure 2 in the textual, Penman style (Penman Natural Language Group, 1989) as opposed to a DAG.

```
(d / drive-01 :mode imperative
    :ARG0 (y / you)
    :ARG1 y
    :destination (b / barrel
        :ARG1-of (l / left-20)))
```

Figure 2: AMR graph for the input *Drive to the barrel on the left* and the alternatively-worded input *Take a drive to the left barrel.*

AMR therefore offers a level of abstraction that is suitable for a robot to act upon as it glosses over some of the linguistic complexity that does not carry any meaningful difference for execution. Furthermore, we find that AMR is well-suited as an input representation to the grounding component because the node concepts of the graph that are grounded are restricted to the action concept and its parameters (such as the destination of a movement instruction). Leveraging AMR allows us to directly associate the **meaning** of the instructions with the physical world, instead of attempting to ground all of the **words** of the instruction, which may include syntactic scaffolding, such as *take* in *take a drive*, that has no grounding in a robot's behavior or the objects in its environment. Benefits of leveraging AMR are further described in §4.1.

After parsing, the wrapper code will interpret the textual representation output from the AMR parser and generate outgoing ROS messages to be published on an established ROS topic. Any ROS software can obtain these messages by subscribing to this topic. In our case, the grounding software component running on the robot will take in these messages and ground the instruction into mission commands for the robot.

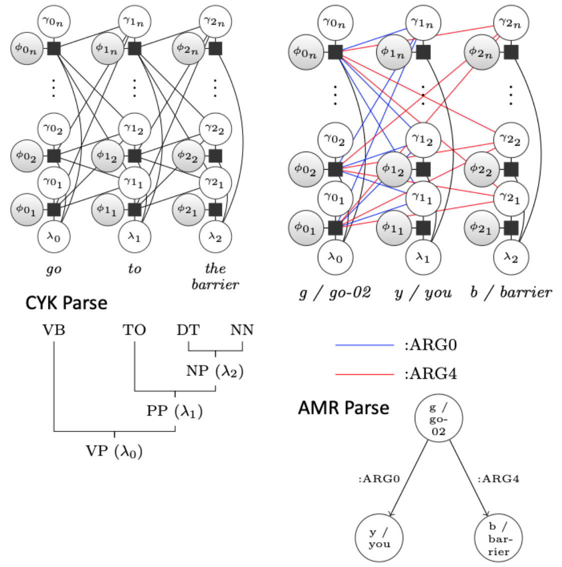

Figure 3: The constituency parse-based DCG on the left exhibits the same number of factors but lacks the informative relational structures of the AMR-based DCG on the right.

3.5 Grounding Component

We take a graphical approach to grounding using a model based on the Distributed Correspondence Graph (Howard et al., 2014). A DCG consists of a set of constituents of language $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$ (e.g., phrases in a parse tree or nodes/edges in an AMR graph), a world model Υ (typically a metric-semantic object-level model), a set of grounding symbols $\Gamma = \{\gamma_1, \ldots \gamma_M\}$ that represent physical concepts (e.g., objects, spatial relationships, robot actions), and a set of binary correspondence variables $\Phi = \{\phi_{11}, \ldots \phi_{NM}\}$ representing True or False correspondence between an individual phrase and individual grounding symbol.

The formulation of DCGs assumes conditional independence of both grounding symbols and linguistic constituents excepting child constituents, resulting in a factor graph hierarchically structured according to the representation of language. Each factor computes the probability of correspondence (ϕ) between a given phrase (λ) and grounding symbol (γ), in the context of a model of the environment. The probabilities are computed by a single log-linear model (Collins, 2005) consisting of expert-designed binary features with associated optimized weights trained from a corpus of annotated data. The features jointly evaluate properties of language and the world, such as a unigram feature for *barrier* and an indicator feature for an object grounding symbol that is True if the object is a *barrier* type, thereby allowing the log-linear model to learn to ground language in physical concepts. Inference is performed via bottom-up beam search to find the most likely `True` correspondences for each linguistic constituent; this process propagates up the hierarchy of the graph. The grounded interpretation of the instruction is represented by the `True` corresponding symbols at the root.

In previous works (Paul et al., 2018; Patki et al., 2020; Howard et al., 2021), a syntactic constituency parse tree, produced by the Cocke-Younger-Kasami (CYK) parsing algorithm (Younger, 1967), was used to represent language instructions; the resulting DCGs inherited the compositional structure of the hierarchy of phrases. A novelty of this work is that we construct a DCG from an AMR parse. A DCG constructed from an AMR parse differs than one constructed from a constituency parse tree because the edges in an AMR parse are labeled. In this work, we assume that there are no cycles in the AMR parse. Consider the example illustrated in Figure 3. For the same language, a parse tree is shown on the left and an AMR parse is on the right. The corresponding constituency parse-based DCG, also shown on the left, expresses a set of symbols for the phrases *the barrier*, *to the barrier*, and *go to the barrier*, where the symbols corresponding to the last phrase represent the grounding of the entire statement. The structure of the AMR-based DCG, shown on the right, differs. Here the AMR-based DCG expresses a set of symbols for the node concepts `y / you`, `b / barrier`, and `g / go-02`. How `y / you` and `b / barrier` are interpreted by `g / go-02` is influenced by the labels of each edge, which are `:ARG0` and `:ARG4`, respectively. To properly capture the structure of this AMR parse, the associated DCG must incorporate the labels of each edge into its own structure; this provides the edge label context to the log-linear model features at each factor, which is necessary to correctly interpret the expressed symbols at child nodes. These differently labeled edges, illustrated in red and blue respectively, are now used in the construction of DCGs so that the engineered features that compose the log-linear model-based factors can utilize this information when determining if a feature is active or inactive. AMR also differs from parse trees in that nodes are permitted to have more than one parent (reentrancy). These are naturally handled by the conditional independence of linguistic constituents that is assumed in the DCG formulation.

In this example, although both models exhibit the same number of factors, the structure of the AMR-based DCG provides richer information, including an explicit representation of who is meant to execute the command. This information is left out of the CYK-based DCG when the imperative is used, as the subject is omitted in the English imperative form.

There are other situations where an AMR-based DCG is preferable to a constituency parse tree-based DCG. For example, the approach leveraging CYK parses required training instances reflecting alternative wordings of what is semantically the same instruction, such as for light-verb constructions. In contrast, our approach enables grounding with less training data since we are grounding the deeper meaning instead of the surface word-forms of the instruction. Another benefit to grounding the meaning behind the instruction, as opposed to the words themselves, is that our implementation is able to more efficiently ground instructions involving co-reference and complex spatial relations, both of which are represented explicitly and consistently in AMR (see §4.1 for further discussion).

3.6 Mission Planner and Executor

Once the action and the action parameters, including any objects mentioned in the instruction, have been grounded, the grounding component sends the action specification to the mission planner. The grounded action includes specifications such as path end points as specified by the location of grounded objects in the robot's world model. For this implementation, we use Cohen et al. (2010)'s Search Based Planning Library global planner and Howard and Kelly (2007)'s Nonlinear Optimization (NLOPT) local planner. Once a plan has been established, the robot mission executor generates and performs the appropriate actions, taking into account real-time feedback from the robot such as the perception of moving obstacles. This completes the loop from natural language instruction to execution within the robot's current physical environment.

4 Demo Description

In the demo, audience members will be invited to interact with the system at a computer workstation

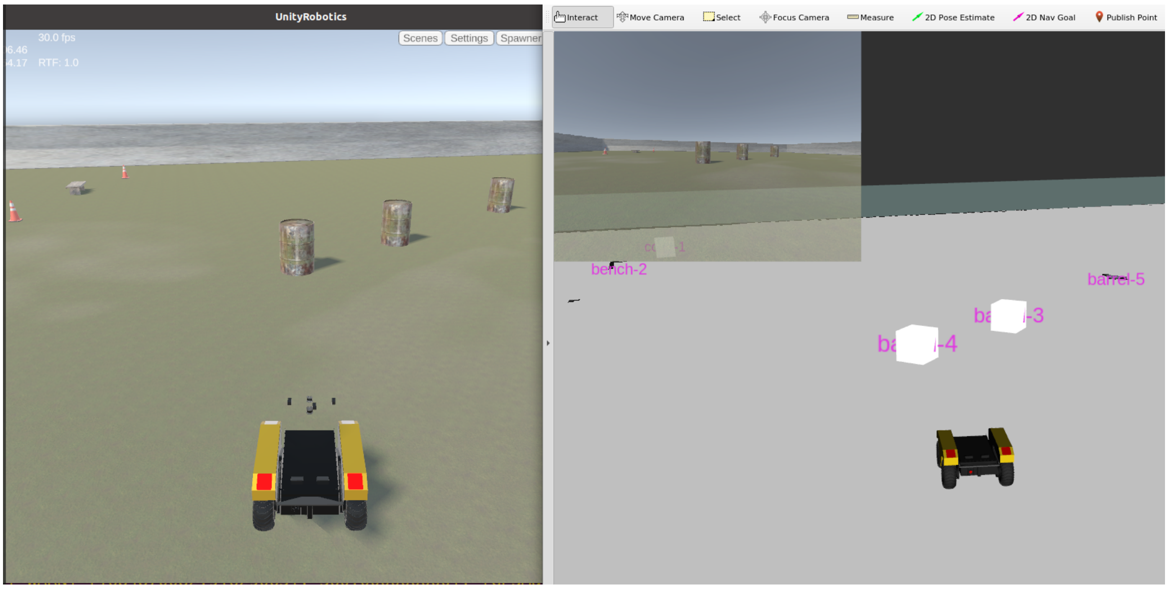

Figure 4: Screen capture of demo, where the left pane of the screen shows the robot's position within the simulation and the right pane of the screen shows the robot's world model view, populated by a LIDAR terrain map and labeled, recognized objects.

where the audience can see a view of the robot in the simulated environment on one side of the screen and what is essentially the robot's view of the world, or its world model, on the other side of the screen. The world model pane shows a LIDAR-derived map of the simulated environment where detected objects in the environment are represented by white boxes with pink labels.

Figure 4 is a screen capture of the demo workstation. In the left pane, visitors can see that the robot is approaching a set of three barrels extending out to the right, and two cones with a small bench in between them ahead of the robot and to its left. In the right pane, visitors can see a visual representation of what the robot "sees" in this same environment using its LIDAR and computer vision sensors. There is a snapshot of exactly what the robot sees from it's onboard camera in the small pane in the top left corner of the right pane. The rest of the right pane populates with light grey in the areas reached by the LIDAR that have been classified as open space; thus, there are some darker grey unknown or unexplored areas beyond the grey barrier that encloses the demo environment. The robot recognizes the three barrels, the cones and the bench. These objects are labeled with the basic object type label as well as a unique identifier number that tracks these objects in the robot's world model. For example, the robot labels the closest barrel as "barrel-4". Demo audience members will be able to direct the robot to any of the objects in the scene that the robot has identified and successfully labeled thus far.

4.1 Demo Modes Comparing AMR & CYK Parsers

In order to showcase the novel contribution of this research, the demo host can toggle the implementation back and forth between the same architecture with either the AMR parser described in Section 3.4, or the syntactic CYK parser (Younger, 1967) of previous implementations, such as Howard et al. (2021). This setup allows us to compare our architecture to comparable systems where the CYK parser was used. However, to make this a fair comparison that focuses only on the language parse and the grounding component, we hold the rest of the architecture constant while only swapping out and comparing the symbol grounding components. This will allow audience members to use different variations of navigation instructions in order to see how a small amount of complexity in the surface form can affect the grounding when using meaning-based (this work) or syntax-based (baseline) parsers.

For example, in our own preliminary comparisons, an experimenter issued the following set of three instructions, given in the same simulated environment to a robot with the same sensors and resulting world model. Only the AMR-based system was able to ground the final two instructions, which involve a light verb construction (2), and coreference as well as a complex spatial expression (3):

1. Go to the left barrel.
2. Take a drive to the left barrel.
3. Drive to the cone and the rock closest to it.

While sufficient for the simple instruction in (1), the syntactic CYK parser output fails to be grounded for instruction (2) because the system cannot ground what it presumes to be the primary *take* action, which has not been seen in training data for either the constituency parse or AMR-based grounding. In the AMR input, *take* is abstracted away and this instruction is grounded to a driving behavior.

For instruction (3), the CYK parser output includes the words *cone* and *it*, which are co-referential expressions for the same object in the environment; thus, the constituency parse tree-based grounding component attempts to separately ground each word. Although this may eventually result in the correct grounding, it is much more computationally expensive and requires a larger space of potential groundings, including symbols for each co-referential expression, in order to compositionally build the grounded meaning of each linguistic constituent. In contrast, AMR represents co-referential expressions as a single node, which is then grounded to a single symbolic meaning.

Instruction (3) also includes the complex spatial expression *the rock closest to it*, which, combined with the coreference, causes the syntax-based grounding to fail altogether. The AMR specifies this as the `close` relation between the concepts of *rock* and *cone*, abstracting away any explicit constituent for the word *it*. Thus, the AMR enables grounding of such spatial concepts to real-world spatial relations between objects in the world model observed in training data.

5 Related Work

This research is at the intersection of NLP, including semantic parsing and dialogue systems, and robotics. We limit our direct comparison here to similarly interdisciplinary work; see Tellex et al. (2020) for a full review of research in robotics and language. Outside of the work on the DCG grounding approach that we directly augment for AMR (Howard et al., 2021), field robotics has largely focused on robots that receive an initial, static tasking and then operate autonomously (e.g., Williams et al. (2012); Arvidson et al. (2010); Camilli et al. (2010)), or robots that are tele-operated (e.g., Kang et al. (2003); Ryu et al. (2004); Yamauchi (2004)). In contrast, there is relatively little work like ours, seeking to develop robots that are able to be tasked dynamically and interactively via natural language.

There are, however, several notable exceptions. Walter et al. (2015) describe the development of a voice-controlled fork lift. In contrast to our own research, however, the natural language instructions are more constrained to particular hard-coded commands mentioning a more limited range of objects that are classified in their world model. Additionally, Heikkilä et al. (2012) develop a mobile manipulator designed for space operations that is capable of accepting spoken commands. Unlike both of the previously mentioned voice-controlled robots, it is important to note that our architecture aims to support bi-directional communication between the robot and the operator, such that ambiguities that might arise in changing environments can be resolved.

There is also relevant research leveraging large, pretrained language models to map or translate between unconstrained natural language and the controlled planning languages of robots. Song et al. (2022) utilize GPT for deciding upon the appropriate high-level plan given natural language instructions, and then use a more traditional low-level planning component to execute specific motor movements to specific grounded points in the environment. The high-level and low-level models are also able to communicate, such that the high-level model can be queried for new and updated plans if conflicts arise in the low-level planning model. Driess et al. (2023) develop their own multi-modal "embodied" language model, called PaLM-E, which accepts both sensor data, such as image data, and natural language text. The model outputs text data that can be interpreted as robot policies. In general, we see potential for leveraging language models in the future both for providing some *apriori*, zero-shot knowledge of objects that the robot might encounter in its environment, which can be used to inform the interpretation of natural language instructions, as well as for providing a likely mapping between unconstrained natural language and the constrained set of robot behaviors.

However, explainability is critical for adoption of robotic systems in high-stakes tasks such as disaster relief; thus, further research enabling transparency and explainability of systems leveraging language models is needed. Neuro-symbolic approaches (e.g., Dipta et al. (2022)) are promising for providing greater transparency. For example, Zhang et al. (2022) develop DANLI, which symbolically represents subgoals as predicates on objects in the robot's world model.

There is a growing body of research leveraging AMR for NLU in human-agent interaction. The present research is part of a broader ongoing research effort leveraging a two-step NLU pipeline that first parses natural language into AMR, which abstracts away from some surface variation, but then in a second step converts the Standard-AMR into a formalism called Dialogue-AMR (Bonial et al., 2020). Dialogue-AMR is augmented to capture features of language found to be critical for human-robot dialogue, but not included in Standard-AMR (Bonial et al., 2019). Specifically, the Dialogue-AMR adds information on the input instruction's tense and aspect, and further normalizes varying expressions for a desired behavior (e.g., *turn, rotate, pivot*) to a single designated roleset for a particular robot behavior (e.g., `turn-01`). While the present research leverages Standard-AMR as the input to the grounding component, we will shift to using Dialogue-AMR as the input parse, as we expect that the further normalization will allow us to achieve comparable results with even less training data. Furthermore, Dialogue-AMR leverages spatial rolesets from Spatial-AMR (Bonn et al., 2020), which provides detailed relations for spatial relations for expressions such as *in front of*, which currently does not have a detailed representation with a relational concept in Standard-AMR.

Other research to augment AMR for interaction includes work to further develop multi-modal, gestural AMR (Brutti et al., 2022) as well as efforts to further develop aspect and modality in AMR to support NLU (Donatelli et al., 2020). Finally, there is research in leveraging AMR parses of image captions in order to develop scene graphs, which can help agents to summarize and process visual scenes (Choi et al., 2022a,b). Together, all of these threads of research demonstrate ways in which AMR can serve as a unified representation for making sense of multiple modalities of information.

6 Conclusions and Future Work

We are currently engaged in experimentation to evaluate the AMR-based grounding. Our ongoing extrinsic evaluation compares natural language interaction with the current paradigm of teleoperation. Specifically, we compare the time it takes for a robot operating autonomously to complete natural language instructions, using the architecture shown in Figure 1, to the time that it takes a relatively experienced person to teleoperate the robot and complete the same instruction. This comparison is made with and without the introduction of latency, which can occur when operators teleoperate a robot from distant, remote locations. The latency, or delay, between the manual teleoperation and the robot's execution of the teleoperation can be disorienting to operators (imagine, for example, if movements of your own body were delayed for some time after your brain sending the signal to move). This disorientation can cause delays in reaching the destination, an inability to reach precise locations, or even crashes. Such latency does not have a dramatic effect on natural language instructions, since although these might be delayed momentarily in getting to the robot, the robot is then navigating autonomously based upon the plan expressed in language. Our early results show that while autonomous navigation is generally slower than teleoperation, with anything over one second of latency introduced, the speed of autonomous navigation becomes comparable.

We are also carrying out an intrinsic evaluation where we compare our architecture, with the AMR parser, against an implementation with a CYK parser in order to robustly evaluate our system against the comparable system of Howard et al. (2021). We will evaluate the performance in terms of the ability of each system to successfully ground a wide variety of instructions with the same training set, and we will also compare computation time and efficiency. Once our evaluations leveraging Standard-AMR are complete, we will then turn to comparing to the use of Dialogue-AMR, where we expect even greater computational efficiency since Dialogue-AMR abstracts even further from surface variation to normalize a variety of different expressions of different behaviors into a single AMR roleset designated for a robot behavior.

Finally, although not the focus of this paper, we are also working to update our architecture such that the intent classification and dialogue management components work more synergistically with the grounding and planning components. Therefore, the system can draw upon its knowledge of the surrounding environment to support more human-like conversational repairs in cases of ambiguities and miscommunications. For example, if the system encounters the well-formed instruction, *Move to the barrel on the right*, but there is no barrel grounded on the right and instead a barrel grounded on the robot's left, then that information from the grounding component can support generation, via AMR, of a targeted clarification question, such as *I don't see a barrel on the right; do you mean the one on the left?* This requires a level of intercommunication of the components that we currently have not achieved.

In this demonstration of our research, we show that AMR-based grounding of natural language instructions allows our system to successfully ground and execute instructions with a range of linguistic phenomena, including light verb constructions, coreference, and spatial relations. Although these phenomena are arguably complex for grounding and have proven to be challenging for the existing state-of-the-art systems, they are commonplace in natural language; thus, we simply must have systems that can handle such complexity reliably in disaster relief scenarios. In the demonstration that we offer, visitors will be able to explore this firsthand to see how our system addresses these challenges by grounding the **meaning** of the instructions, rather than just the **words** of the instructions.

References

Raymond E Arvidson, James F Bell III, P Bellutta, Nathalie A Cabrol, JG Catalano, J Cohen, Larry S Crumpler, DJ Des Marais, TA Estlin, WH Farrand, et al. 2010. Spirit mars rover mission: Overview and selected results from the northern home plate winter haven to the side of scamander crater. *Journal of Geophysical Research: Planets*, 115(E7).

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Claire Bonial, Lucia Donatelli, Mitchell Abrams, Stephanie M. Lukin, Stephen Tratz, Matthew Marge, Ron Artstein, David Traum, and Clare Voss. 2020. Dialogue-AMR: Abstract Meaning Representation for dialogue. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 684–695, Marseille, France. European Language Resources Association.

Claire Bonial, Lucia Donatelli, Stephanie M. Lukin, Stephen Tratz, Ron Artstein, David Traum, and Clare Voss. 2019. Augmenting Abstract Meaning Representation for human-robot dialogue. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 199–210, Florence, Italy. Association for Computational Linguistics.

Julia Bonn, Martha Palmer, Zheng Cai, and Kristin Wright-Bettner. 2020. Spatial AMR: Expanded spatial annotation in the context of a grounded Minecraft corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4883–4892, Marseille, France. European Language Resources Association.

Richard Brutti, Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2022. Abstract meaning representation for gesture. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1576–1583.

Richard Camilli, Christopher M Reddy, Dana R Yoerger, Benjamin AS Van Mooy, Michael V Jakuba, James C Kinsey, Cameron P McIntyre, Sean P Sylva, and James V Maloney. 2010. Tracking hydrocarbon plume transport and biodegradation at deepwater horizon. *Science*, 330(6001):201–204.

Woo Suk Choi, Yu-Jung Heo, Dharani Punithan, and Byoung-Tak Zhang. 2022a. Scene graph parsing via abstract meaning representation in pre-trained language models. In *Proceedings of the 2nd Workshop on Deep Learning on Graphs for Natural Language Processing (DLG4NLP 2022)*, pages 30–35.

Woo Suk Choi, Yu-Jung Heo, and Byoung-Tak Zhang. 2022b. Sgram: Improving scene graph parsing via abstract meaning representation. *arXiv preprint arXiv:2210.08675*.

Clearpath Robotics. 2023. Clearpath Husky UGV.

Benjamin J Cohen, Sachin Chitta, and Maxim Likhachev. 2010. Search-based planning for manipulation with motion primitives. In *2010 IEEE international conference on robotics and automation*, pages 2902–2908. IEEE.

Michael Collins. 2005. Log-linear models. *Self-Published Tutorial*.

Shubhashis Roy Dipta, Mehdi Rezaee, and Francis Ferraro. 2022. Semantically-informed hierarchical event modeling.

Lucia Donatelli, Kenneth Lai, and James Pustejovsky. 2020. A two-level interpretation of modality in human-robot dialogue. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4222–4238, Barcelona, Spain (Online).

International Committee on Computational Linguistics.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. Palm-e: An embodied multimodal language model.

Arno Hartholt, David Traum, Stacy C Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. 2013. All together now: Introducing the virtual human toolkit. In *Intelligent Virtual Agents: 13th International Conference, IVA 2013, Edinburgh, UK, August 29-31, 2013. Proceedings 13*, pages 368–381. Springer.

Seppo S Heikkilä, Aarne Halme, and André Schiele. 2012. Affordance-based indirect task communication for astronaut-robot cooperation. *Journal of field robotics*, 29(4):576–600.

Thomas Howard and Alonzo Kelly. 2007. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, 26(2):141 – 166.

Thomas M Howard, Nicholas Roy, Jonathan Fink, Jacob Arkin, Rohan Paul, Daehyung Park, Subhro Roy, D Barber, Rhyse Bendell, Karl Schmeckpeper, et al. 2021. An intelligence architecture for grounded language communication with field robots. In *Field Robotics, 2021*. Field Robotics.

Thomas M. Howard, Stefanie Tellex, and Nicholas Roy. 2014. A natural language planner interface for mobile manipulators. pages 6652–6659. Institute of Electrical and Electronics Engineers Inc.

Sungchul Kang, Changhyun Cho, Jonghwa Lee, Dongseok Ryu, Changwoo Park, Kyung-Chul Shin, and Munsang Kim. 2003. Robhaz-dt2: Design and integration of passive double tracked mobile manipulator system for explosive ordnance disposal. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2624–2629. IEEE.

Anton Leuski and David Traum. 2011. Npceditor: Creating virtual human dialogue using information retrieval techniques. *Ai Magazine*, 32(2):42–56.

Matthias Lindemann, Jonas Groschwitz, and Alexander Koller. 2019. Compositional semantic parsing across graphbanks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4576–4585, Florence, Italy. Association for Computational Linguistics.

Matthew Marge, Claire Bonial, Brendan Byrne, Taylor Cassidy, A William Evans, Susan G Hill, and Clare Voss. 2016. Applying the Wizard-Of-Oz Technique to Multimodal Human-Robot Dialogue. In *Proc. of IEEE International Symposium on Robot and Human Interactive Communication*.

Robin R Murphy. 2014. *Disaster robotics*. MIT press.

Siddharth Patki, Ethan Fahnestock, Thomas M Howard, and Matthew R Walter. 2020. Language-guided semantic mapping and mobile manipulation in partially observable environments. In *Conference on Robot Learning*, pages 1201–1210. PMLR.

Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M. Howard. 2018. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *International Journal of Robotics Research*, 37:1269–1299.

Penman Natural Language Group. 1989. The Penman user guide. *Technical report, Information Sciences Institute*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society. IEEE Catalog No.: CFP11SRW-USB.

Dongseok Ryu, Sungchul Kang, Munsang Kim, and Jae-Bok Song. 2004. Multi-modal user interface for teleoperation of robhaz-dt2 field robot system. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 1, pages 168–173. IEEE.

Chan Hee Song, Jihyung Kil, Tai-Yu Pan, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2022. One step at a time: Long-horizon vision-and-language navigation with milestones. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15482–15491.

Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:25–55.

David Traum, Cassidy Henry, Stephanie Lukin, Ron Artstein, Felix Gervits, Kimberly Pollard, Claire Bonial, Su Lei, Clare Voss, Matthew Marge, et al. 2018. Dialogue structure annotation for multi-floor interaction. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Matthew R Walter, Matthew Antone, Ekapol Chuangsuwanich, Andrew Correa, Randall Davis, Luke Fletcher, Emilio Frazzoli, Yuli Friedman, James Glass, Jonathan P How, et al. 2015. A situationally aware voice-commandable robotic forklift working alongside people in unstructured outdoor environments. *Journal of Field Robotics*, 32(4):590–628.

Stefan B Williams, Oscar R Pizarro, Michael V Jakuba, Craig R Johnson, Neville S Barrett, Russell C Babcock, Gary A Kendrick, Peter D Steinberg, Andrew J Heyward, Peter J Doherty, et al. 2012. Monitoring of benthic reference sites: using an autonomous underwater vehicle. *IEEE Robotics & Automation Magazine*, 19(1):73–84.

Brian M Yamauchi. 2004. Packbot: a versatile platform for military robotics. In *Unmanned ground vehicle technology VI*, volume 5422, pages 228–237. SPIE.

Daniel H Younger. 1967. Recognition and parsing of context-free languages in time n3. *Information and control*, 10(2):189–208.

Yichi Zhang, Jianing Yang, Jiayi Pan, Shane Storks, Nikhil Devraj, Ziqiao Ma, Keunwoo Peter Yu, Yuwei Bao, and Joyce Chai. 2022. Danli: Deliberative agent for following natural language instructions. *arXiv preprint arXiv:2210.12485*.