Experiments in Proactive Symbol Grounding for Efficient Physically Situated Human-Robot Dialogue

Jacob Arkin
Robotics and Artificial Intelligence Lab
Electrical and Computer Engineering
University of Rochester
Rochester, NY 14623, USA
j.arkin@rochester.edu

Thomas M. Howard
Robotics and Artificial Intelligence Lab
Electrical and Computer Engineering
University of Rochester
Rochester, NY 14623, USA
thoward@ece.rochester.edu

Abstract

Real-time performance of human-robot dialogue is important for making the interaction effective and worthwhile. Contemporary approaches treat language understanding and generation as reactive processes that construct a new inference or inverse-semantics model upon receiving a novel utterance. In this work, we consider the proactive generation and symbol grounding of likely relevant utterances as a means of improving the computational efficiency of dialogue interaction. We explore this approach in the navigation domain as applied to a human teammate providing navigation commands to an unmanned ground vehicle.

1 INTRODUCTION

Effective human-robot teams that operate in complex, dynamic, and uncertain environments must share common representations of objects and other spatial concepts used in the specification of tasks that each may perform. Recent progress in natural language understanding (NLU) and generation has led to increasingly sophisticated algorithms that infer distributions of symbols representing physical meaning or ask questions to clarify ambiguity.

A limitation of NLU approaches for collaborative robots is that they react to inputs rather than predict and precompute solutions for future interactions. Consider the scenario illustrated in Figure 1, where an unmanned ground vehicle has encountered a scene with many different types of semantic objects. A system that has proactively grounded phrases that uniquely describe the different objects in the scene (e.g., "the green ball on the left") can bootstrap the probabilistic inference with partial or complete solutions.

Figure 1: A scenario where we may require real-time human-robot dialog.

We present a method for proactive symbol grounding (PSG) in the context of NLU and explore its use for following robot instructions and clarifying ambiguous responses via question generation using inverse semantics. The PSG is made efficient by exploiting both Distributed Correspondence Graphs (DCGs) (Howard et al., 2014) for probabilistic inference and a bottom-up algorithm for sampling candidate phrases.

2 BACKGROUND

The problem of providing robots with the capacity to engage in dialogue can be considered to have two main components: language understanding of user-provided utterances and language generation of responses for the robot to express to the user.

2.1 Language Understanding for Robots

Some research has leveraged rule-based approaches for a variety of applications (Dzifcak et al., 2009; Kruijff et al., 2007). Other efforts have emphasized probabilistic approaches that learn models of the association between language and robot actions or paths in the world (Vogel and Jurafsky, 2010; Matuszek et al., 2010).

More recently, some approaches have posed the problem as inference over a factor graph structured according to the constituency parse of language (Tellex et al., 2011; Paul et al., 2016).

A unifying characteristic of these works is their reactive nature, waiting to receive an utterance before inferring any meaning. The approach presented in this paper proactively generates a space of language unprompted and infers meaning for samples within that space to be used as partial or complete solutions for a novel utterance.

2.2 Language Generation for Dialogue

Traditional approaches for language generation emphasize sentence planning elements such as surface realization (Chen and Mooney, 2011; Garoufi and Koller, 2011). However, in the context of physically situated dialogue, it is necessary to reason about environmental context. We are interested in approaches that generate language by inverting the NLU process.

Our work takes inspiration from Knepper et al. (2015) in that we choose to invert a probabilistic NLU model that uses environmental context during inference. It thus becomes possible to generate unambiguous language descriptions of particular concepts. We generate a space of language from a provided grammar and precompute the meaning of phrase samples in order to bootstrap the language generation process during interaction.

3 TECHNICAL APPROACH

Our formulation of NLU assumes conditional independence across linguistic and symbolic constituents to efficiently infer a distribution of symbols for an utterance. As used in equation 1, Γ represents the space of symbolic constituents, Λ represents a constituency parse tree, Φ defines a space of correspondence variables, Φ_{ci} are the child phrase correspondence variables, and Υ is the assumed environment model. For a more complete description, we refer to Howard et al. (2014).

$$\Phi^* = \underset{\phi_{ij} \in \Phi}{\arg\max} \prod_{i=1}^{|\Lambda|} \prod_{j=1}^{|\Gamma|} p\left(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{ci}, \Upsilon\right) \quad (1)$$

An overview of the architecture is illustrated in Figure 2. The PSG module accepts both a grammar definition and a list of perceived objects. It publishes updates of proactively grounded phrases to the natural language symbol grounding (NLSG) module, which also accepts the same list of perceived objects. The parser module accepts a text-based instruction and constructs a parse tree from the grammar to send to the NLSG module. The NSLG module searches that tree for subphrases matching known proactively grounded phrases and copies the symbols, thus eliminating those phrases from online inference. Once inference completes, a list of actions is extracted from the root of the instruction and processed by either a motion planner when the action is understood or by a dialog system when the action is ambiguous.

Mathematically, we consider PSG to reduce the number of phrases that we need to evaluate for a novel instruction, defined as a reduction in phrases $\hat{\Lambda}$ as a function of the original parse tree Λ and the proactively grounded phrases $\mathcal{PSG}$:

$$\hat{\Lambda} = f\left(\Lambda, \mathcal{PSG}\right) \quad (2)$$

We modify Equation 1 to use $\hat{\Lambda}$.

$$\Phi^* = \underset{\phi_{ij} \in \Phi}{\arg\max} \prod_{i=1}^{|\hat{\Lambda}|} \prod_{j=1}^{|\Gamma|} p\left(\phi_{ij}, \gamma_{ij}, \lambda_i, \Phi_{ci}, \Upsilon\right) \quad (3)$$

The problem of inferring the proactively grounded phrases $\mathcal{PSG}$ is one of predicting and proactively grounding salient phrases that represent physical or abstract concepts in the environment. We propose a bottom-up approach to searching the grammar for relevant phrases that first constructs a space of phrases that are only composed of part-of-speech tags. The PSG module randomly samples from this space and uses the DCG to infer the meaning of each sample. The solutions are stored and published to the NLSG module for use during inference over a user's novel instruction. Once the space of candidate phrases has exhausted, new candidates are constructed by searching the grammar for rules that contain phrase types matching the roots of existing candidates. The process repeats until candidates reach a specified depth and/or the environment changes sufficiently to invalidate solutions.

The next two sections describe our method for evaluating the performance of the proposed algorithm and present preliminary results of PSG for improving the run-time performance of both NLU and clarifying-response generation.

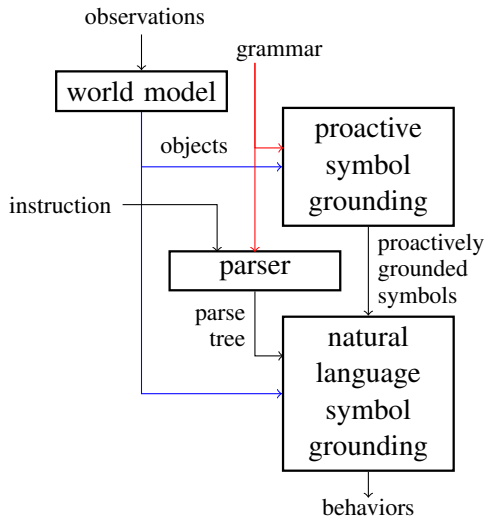

Figure 2: The proposed PSG system architecture.

4 EXPERIMENTAL DESIGN

To evaluate the performance and applicability of PSG, we propose two quantitative experiments focusing on runtime for language understanding and generation. We expect the impact of PSG to depend on the amount of time spent constructing and grounding candidate phrases; thus, the first experiment evaluates the number of grounded candidate phrases and associated novel inference runtime for increasing durations of time spent grounding candidate phrases. The second experiment is identical except we observe the runtime for generating a disambiguating query using inverse semantics rather than the runtime for language understanding. Both experiments assume a symbolic representation, grammar, and corpus of annotated examples used to train the DCG.

The corpus used in both experiments consists of thirty-four navigation commands (e.g. "navigate to the farthest cone on the right in the row of cones") consistent with the examples presented in Paul et al. (2016). The commands are associated with three different simulated worlds composed of differently colored balls and cones.

We can also extract a grammar that is used by the parser and defines the space of candidate phrases for PSG. The grammar contains thirty individual words and nine phrase rules.

5 RESULTS

5.1 Example Inference

In the first part of our analysis, we consider the impact of PSG on inference time for the phrase "go to the blue ball on the right". The parse tree for this instruction is illustrated in Figure 3.

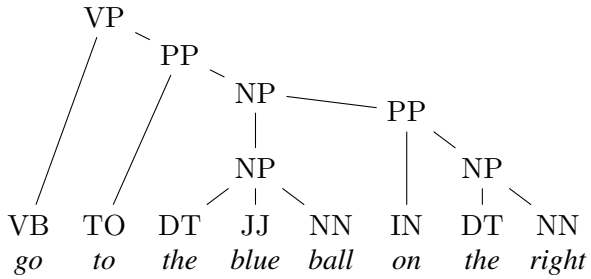

Figure 3: The parse tree for the instruction "go to the blue ball on the right"

To quantify the impact of PSG, we consider the runtime of NLU for increasing durations of time spent grounding candidate phrases, (see Table 1). As the number of grounded candidate phrases increases with longer durations, the likelihood of producing relevant solutions for a novel instruction also increases, thus reducing the number of phrases to evaluate at inference time. In the case where PSG ran for 8 seconds, it provided NLU with solutions for "the blue ball" and "on the right", leaving only three phrases to be evaluated.

PSG (sec)	0.0	2.0	4.0	6.0	8.0
Candidates	0	31	62	102	146
NLU (sec)	0.21	0.18	0.14	0.13	0.09

Table 1: The duration of time spent grounding candidate phrases (PSG) versus inference runtime (NLU) for "go to the blue ball on the right'.

5.2 Inverse Semantics

To demonstrate our approach to inverse semantics, we consider the instruction "navigate to the blue ball" in a cluttered ball/cone environment with two blue balls of varying distance from the robot. The most likely solution for this expression yielded no symbols for the root, only providing symbols indicating object type and color for the phrase "the blue ball". We can search the world model for objects that contain those properties (the two blue balls in this case). To ask a clarifying response, we can search for phrases whose root meanings align with unambiguous labels for either ball. This search process is effectively targeted PSG that terminates when the unambiguous phrases are found; thus, we can boostrap this process using the known cadidate phrases produced by PSG.

After search, we find that the expressions "the nearest blue ball" and "the farthest blue ball" unambiguously describe the blue balls. We can thus generate a template-based query using the command to ask "navigate to the farthest blue ball or navigate to the nearest blue ball?". We consider the runtime of inverse semantics (IS) for increasing durations of time spent grounding candidate phrases (PSG) (see Table 2). As the number of grounded candidates produced increases, the number of phrases IS must evaluate decreases. In the case where PSG ran for 10 seconds, IS was instanteous because the required unambigous phrases existed in the PSG-produced candidate phrases.

PSG (sec)	0.0	2.0	4.0	6.0	8.0	10.0
IS (sec)	9.0	7.4	5.4	3.4	1.5	0.0

Table 2: The duration of time spent grounding candidate phrases (PSG) versus inverse semantics runtime (IS) for generating clarifying dialog for the instruction "navigate to the blue ball'.

6 DISCUSSION/CONCLUSION

The work presented here can be extended to improve computational efficiency in several ways. Prior interactions may inform the likelihood of candidate phrases. Consider a human operator providing an instruction "pick up the ball near the truck". It is reasonable to expect the next instruction will be related, for example "place it in the back of the truck". Modeling the likelihood of candidate phrases based on past dialog interactions is expected to outperform random sampling.

One challenge of the proposed approach is deciding when the environment has sufficiently changed to invalidate precomputed solutions to candidate phrases. A model that efficiently determines the subset of grounded candidate phrases that have been invalidated by the change in the environment would outperform our naive approach of treating the full set as invalid.

In this paper we present a framework for improving the runtime performance of physically situated human-robot dialog via PSG. Experimental results demonstrate improved runtime performance for NLU and for inverse semantics when generating clarifying responses. In future work we aim to study the topics described above and apply our approach to physical platforms.

7 ACKNOWLEDGMENTS

This work was supported in part by the Robotics Consortium of the U.S Army Research Laboratory under the Collaborative Technology Alliance Program

References

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. *San Francisco, CA*, pages 859–865.

Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. 2009. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 4163–4168. IEEE.

Konstantina Garoufi and Alexander Koller. 2011. Combining symbolic and corpus-based approaches for the generation of successful referring expressions. In *Proceedings of the 13th European workshop on natural language generation*, pages 121–131. Association for Computational Linguistics.

Thomas M. Howard, Stefanie Tellex, and Nicholas Roy. 2014. A natural language planner interface for mobile manipulators. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, pages 6652–6659. IEEE.

Ross A Knepper, Stefanie Tellex, Adrian Li, Nicholas Roy, and Daniela Rus. 2015. Recovering from failure by asking for help. *Autonomous Robots*, 39(3):347–362.

Geert-Jan M Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I Christensen. 2007. Situated dialogue and spatial organization: What, where... and why. *Int'l J. of Advanced Robotic Systems*, 4(2):125–138.

Cynthia Matuszek, Dieter Fox, and Karl Koscher. 2010. Following directions using statistical machine translation. In *Proc. ACM/IEEE Int'l. Conf. on Human-Robot Interaction (HRI)*, pages 251–258. IEEE Press.

Rohan Paul, Jacob Arkin, Nicholas Roy, and Thomas M. Howard. 2016. Efficient Grounding of Abstract Spatial Concepts for Natural Language Interaction with Robot Manipulators. In *Proc. Robotics: Science and Systems (RSS)*.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. 2011. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *Proc. Nat'l Conf. on Artificial Intelligence (AAAI)*.

Adam Vogel and Dan Jurafsky. 2010. Learning to follow navigational directions. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 806–814. Association for Computational Linguistics.