Grounding Abstract Spatial Concepts for Language Interaction with Robots

Rohan Paul[1] and **Jacob Arkin**[2] and **Nicholas Roy**[1] and **Thomas Howard**[2]
[1]Massachusetts Institute of Technology, USA
[2]University of Rochester, USA
{rohanp,nickroy}@csail.mit.edu, jarkin@ur.rochester.edu and thoward@ece.rochester.edu

Abstract

Our goal is to develop models that allow a robot to understand or "ground" natural language instructions in the context of its world model. Contemporary approaches estimate correspondences between an instruction and possible candidate groundings such as objects, regions and goals for a robot's action. However, these approaches are unable to reason about abstract or hierarchical concepts such as rows, columns and groups that are relevant in a manipulation domain, Figure 1. We introduce a probabilistic model that incorporates an expressive space of abstract spatial concepts as well as notions of cardinality and ordinality. Abstract concepts are introduced as explicit hierarchical symbols correlated with concrete groundings. Crucially, the abstract groundings form a Markov boundary over concrete groundings, effectively de-correlating them from the remaining variables in the graph which reduces the complexity of training and inference in the model. Empirical evaluation demonstrates accurate grounding of abstract concepts embedded in complex natural language instructions commanding a robot manipulator. The proposed inference method leads to significant efficiency gains compared to the baseline, with minimal trade-off in accuracy.

1 Introduction

Natural language communication with robots has been a long standing goal in robotics and AI. As robots to enter our factories, workplaces and homes where effective communication between humans and robots is vital. Natural language provides a rich, intuitive and flexible medium for humans and robots to interact and share information. The problem of "grounding" or understanding natural language instructions involves determining the higher-order semantic concepts expressed in the utterance and relating these concepts with entities perceived in the world. Inspired by statistical machine translation, recent approaches [Howard *et al.*, 2014b; Tellex *et al.*, 2011; Boularias *et al.*, 2015; Matuszek *et al.*, 2012; Oh *et al.*, 2015] pose the grounding problem as inference on a graphical model structured according to the well studied parse structure of language. These models estimate correspondences between linguistic constituents in an instructions and semantic entities perceived in the world. Present models can only ground concrete entities such as objects/regions, and cannot model abstract semantic concepts. E.g., spatial aggregations like "rows, columns or groups" common in human language, Figure 1.

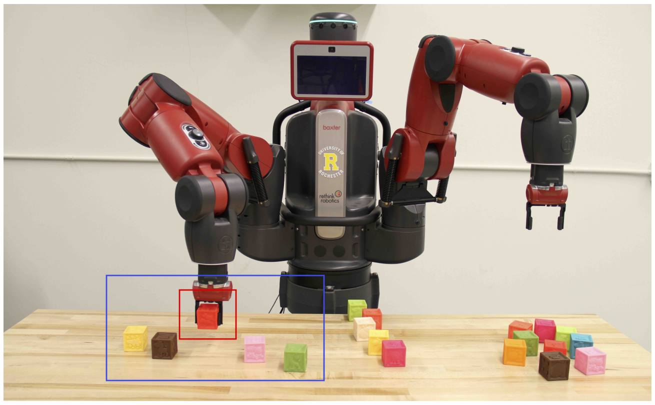
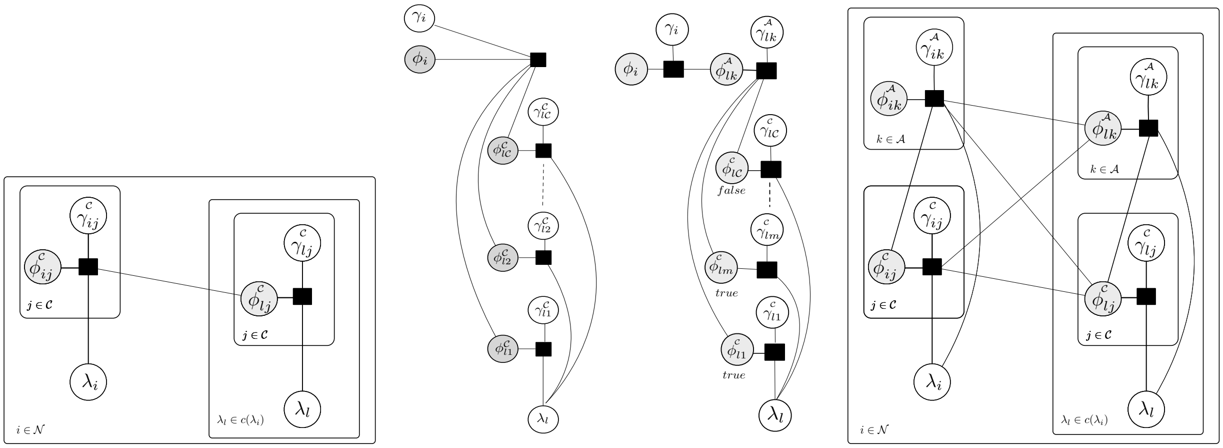


Figure 1: Robot following the instruction, "pick up the middle block in the row of five blocks on the right". The grounding for an aggregative concept ("rows") is abstract and linked with the expression of constituent concrete groundings ("blocks"). The space of abstract concepts is exponentially-large in the number of constituents, 17.3×10^6 symbols in this setup. We present a probabilistic model to efficiently ground abstract concepts in natural language instructions.

We present the Adaptive Distributed Correspondence Graph (ADCG) model that enables grounding of abstract concepts referenced in language utterances. The model introduces a factorization over concrete and abstract symbols such that reflects the hierarchical structure of abstractions and allows efficient approximate inference in the exponentially-large space of abstractions. Empirical evaluation revealed significantly lower inference runtime with equivalent accuracy compared to the state of the art baseline. This work appeared in [Paul *et al.*, 2016]. Here, we present an abridged version elucidating the central contribution.


$\gamma_{ij}^{\mathcal{C}}$
$\phi_{ij}^{\mathcal{C}}$
$j \in \mathcal{C}$
$\gamma_{lj}^{\mathcal{C}}$
$\phi_{lj}^{\mathcal{C}}$
λ_i
λ_l
$i \in \mathcal{N}$
$\lambda_l \in c(\lambda_i)$
γ_i
ϕ_i
$\gamma_{lc}^{\mathcal{C}}$
$\phi_{lc}^{\mathcal{C}}$
$\gamma_{l2}^{\mathcal{C}}$
$\phi_{l2}^{\mathcal{C}}$
$\gamma_{l1}^{\mathcal{C}}$
$\phi_{l1}^{\mathcal{C}}$
$\gamma_{lk}^{\mathcal{A}}$
$\phi_{lk}^{\mathcal{A}}$
$\gamma_{lm}^{\mathcal{C}}$
$\phi_{lm}^{\mathcal{C}}$
false
true
true
$\gamma_{ik}^{\mathcal{A}}$
$\phi_{ik}^{\mathcal{A}}$
$k \in \mathcal{A}$

(a) DCG Model

(b) Introducing Abstractions

(c) Proposed ADCG Model

Figure 2: Factor graph representations. Input instruction is parsed into ($\mathcal{N}$) phrases where (λ_l) represents a child phrase for parent (λ_i). Superscripts ($\mathcal{C}$) and ($\mathcal{A}$) denote concrete and abstract variables. Unknown variable nodes appear in grey. (a) The DCG model includes concrete correspondences ("objects", "regions", "actions" etc.) conditioned on child correspondences relating phrases with groundings. (b) Modeling abstract concepts. (Left) introduction of a joint factor between concrete correspondences ($\phi^{\mathcal{C}}$) expresses all possible abstractions but fully correlates parent groundings. (Right) Introduction of explicit abstract correspondence variables form a Markov boundary over concrete groundings decoupling them from remaining variables. (c) The proposed ADCG model introduces factors for abstract concepts ("rows", "columns", "groups" etc.) that are hierarchically linked with concrete groundings and correspondences from child phrases.

2 Grounding Natural Language Instructions

Consider the robot manipulator operating in a workspace Υ modeled as a collection of rigid bodies. Let, groundings Γ denote semantic concepts that are conveyed by an input language instruction $\mathbf{\Lambda}$. These include notions such as objects, locations, regions derived from the world model or future actions the robot can execute. E.g., the instruction "pick up the block on the table" include objects for phrases "the block" and "the table", the phrase "on" is interpreted as a region above the "table" and the phrase "pick up" associated with the intended grasp action to be executed by the robot. The instruction $\mathbf{\Lambda}$ consists of a set of phrases $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ determined from a parse tree $\tau(\mathbf{\Lambda})$. The grounding problem is posed as estimating the likely set of groundings $\mathbf{\Gamma} = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ for each phrase in the input instruction:

$$\underset{\gamma_1 \ldots \gamma_n \in \mathbf{\Gamma}}{\arg\max} \; p(\mathbf{\Gamma}|\mathbf{\Lambda}, \Upsilon). \tag{1}$$

Contemporary techniques such as Distributed Correspondence Graphs (DCG) [Howard *et al.*, 2014a] pose the grounding problem as inference on a factor graph. The grounding process is mediated by a binary correspondence variable ϕ_{ij} that expresses the degree to which the phrase $\lambda_i \in \mathbf{\Lambda}$ corresponds to a possible grounding $\gamma_{ij} \in \Gamma$. The graph is constructed according to the parse structure of the input instruction such that the latent grounding for a phrase λ_i is conditioned only on groundings $\mathbf{\Phi}_{c_i}$ for child phrases:

$$\underset{\phi_{ij} \in \mathbf{\Phi}}{\arg\max} \; \prod_{i=1}^{|\mathcal{N}|} \prod_{j=1}^{|\mathcal{C}|} p(\phi_{ij}|\gamma_{ij}, \lambda_i, \mathbf{\Phi}_{c_i}, \Upsilon). \tag{2}$$

Next, we introduce Adaptive Distributed Correspondence Graphs (ADCG) that enable modeling and inference over abstract concepts.

3 Probabilistic Model

In this section, we first define the space of groundings followed by detailing the probabilistic model and finally discuss the inference procedure.

3.1 Generalized Space of Groundings

We define the space of grounding symbols Γ that represent semantic concepts expressed in input language $\mathbf{\Lambda}$. The robot's workspace is represented as the set of objects O each possessing geometric, appearance and pose information, typically obtained from a perception system. We assume that the robot is capable of executing a set of manipulation actions ("pick", "place", "clear" etc.) parameterized by the objects under consideration. Additionally, we incorporate symbols expressing cardinality ("two", "three", "four" etc.), ordinality ("fifth", "sixth", "seventh" etc.) and spatial regions ("left", "center", "behind" etc.) associated with objects in the scene. The introduced symbols collectively form the space of concrete symbols $\Gamma^{\mathcal{C}}$.

Abstract concepts such as "rows", "columns", "groups", "towers" etc. are introduced as hierarchical symbols composed of concrete entities as conveyed in an instruction like "the column of red blocks". Formally, an abstraction η includes a subset of objects $O_j \subseteq O$ possessing a common spatial characteristic ("linearity", "circularity", "directivity" etc.) denoted by the set Σ: $\eta = \{(\sigma_i, O_j) | \sigma_i \in \Sigma, O_j \subseteq O\}$. The number of possible containers is $|\Sigma| \times |\mathcal{P}(O)|$, where $\mathcal{P}$ denotes the power set. Hence, the symbol space of containers and associated regions is exponential $O(2^{N_O})$ in the number of objects populating the world model. The symbolic representation presented above forms the space of abstract groundings $\Gamma^{\mathcal{A}}$. Note that the space of concrete groundings grows linearly as $O(N_O)$. However, the space of abstractions is exponentially large in the number of concrete groundings $O(2^{N_O})$. Even in a simplistic block world setup with 20 objects, Figure 1, results in an abstract search space that includes 17.3 million symbols.

3.2 Factor Graph

Estimating the likely correspondences Φ between the input natural language instruction $\mathbf{\Lambda}$ and the generalized space of groundings $\Gamma^{C} \cup \Gamma^{\mathcal{A}}$ is posed as inference on a factor graph. Following the DCG factor graph construction [Howard *et al.*, 2014a], the concrete correspondence variables ϕ^{C}_{ij} are introduced that relate a phrase λ_i with concrete groundings γ^{C}_{ij}. Next, we introduce notions of abstract symbols in the model. Figure 2(b-left) illustrates an approach that introduces a shared factor between all concrete correspondences, modeling an abstract such as a "row" composed of concrete groundings like "blocks". Although, this factor expresses the distribution over possible aggregations of concrete groundings, the representation possesses the disadvantage that grounding for a parent phrase such as, "the middle block in the row" becomes correlated with the inference over abstractions from concrete groundings. Consequently, we introduce explicit abstract grounding variables $\gamma^{\mathcal{A}}_{ik}$ and correspondences $\phi^{\mathcal{A}}_{ik}$, Figure 2(b-right). In this formulation, the abstract correspondences act as indicator functions distinguishing each expressed aggregation and and form a Markov boundary for the concrete correspondences. Given the knowledge of the abstract correspondence variables, a hierarchically linked grounding, like "the middle block" is de-correlated with the joint distribution over concrete constituents. Figure 2(b) presents the factor graph. The joint distribution is modeled as a product of factor potentials Ψ given as:

$$\begin{aligned}\underset{\phi^{C}_{ij},\phi^{\mathcal{A}}_{ik}\in\mathbf{\Phi}}{\arg\max}\Bigg\{ \prod_{i=1}^{|\mathcal{N}|} \Bigg(\prod_{j=1}^{|C|} \Psi(\phi^{C}_{ij}, \gamma^{C}_{ij}, \lambda_i, \{\mathbf{\Phi}^{C}_{c_i} \cup \mathbf{\Phi}^{\mathcal{A}}_{c_i}\}, \Upsilon) \\ \prod_{k=1}^{|\mathcal{A}|} \Psi(\phi^{\mathcal{A}}_{ik}, \gamma^{\mathcal{A}}_{ik}, \lambda_i, \{\mathbf{\Phi}^{C}_{i} \cup \mathbf{\Phi}^{C}_{c_i} \cup \mathbf{\Phi}^{\mathcal{A}}_{c_i}\}, \Upsilon)\Bigg)\Bigg\}.\end{aligned} \tag{3}$$

Each factor in the model relates unknown concrete ϕ^{C}_{ij} or abstract $\phi^{\mathcal{A}}_{ik}$ correspondences between an input phrase λ_i and probable groundings γ^{C}_{ij} or $\gamma^{\mathcal{A}}_{ik}$ respectively. The grounding for a phrase is conditioned on expressed groundings from the child phrase indicated by correspondences $\mathbf{\Phi}^{C}_{c_i}$ and $\mathbf{\Phi}^{\mathcal{A}}_{c_i}$. An abstract factor links the input phrase with correspondences related to abstract groundings. Let the variable set $\mathbf{\Phi}^{C}_{i}$ denote the concrete correspondences for the current phrase. Each abstract factor estimates correspondence $\phi^{\mathcal{A}}_{ik}$ and jointly reasons with the estimated concrete child groundings $\mathbf{\Phi}^{C}_{i}$ for the current phrase. This expresses the probabilistic linkage between a hierarchical abstract grounding and the set of concrete groundings, e.g. "column of blocks on the right". Factor potentials Ψ are endowed with a log-linear model composed of a linear combination of predictive feature function incorporating spatial and lexical cues.

3.3 Approximate Inference

Inference in the graphical model is posed as tree-structured search over possible correspondences between phrases and groundings given child context. The inclusion of abstract groundings leads to an exponential increase in the size of the search space, rendering exhaustive search infeasible. In order to search efficiently, we leverage a partitioning of the joint distribution between concrete and abstract factors, Equation 3. For each phrase, factor evaluations are ordered such that the distribution over concrete symbols given child groundings is determined first and the set of probable solutions above a confidence threshold are obtained. The set of possible object groundings are obtained as:

$$\widehat{O(i)} = \{\widehat{o_j} | \gamma^{C}_{ij} = \widehat{o_j}, p(\phi^{C}_{ij} | \gamma^{C}_{ij}, \lambda_i, \mathbf{\Phi}_{\boldsymbol{c_i}}) \geq p_T, j \in C\}. \tag{4}$$

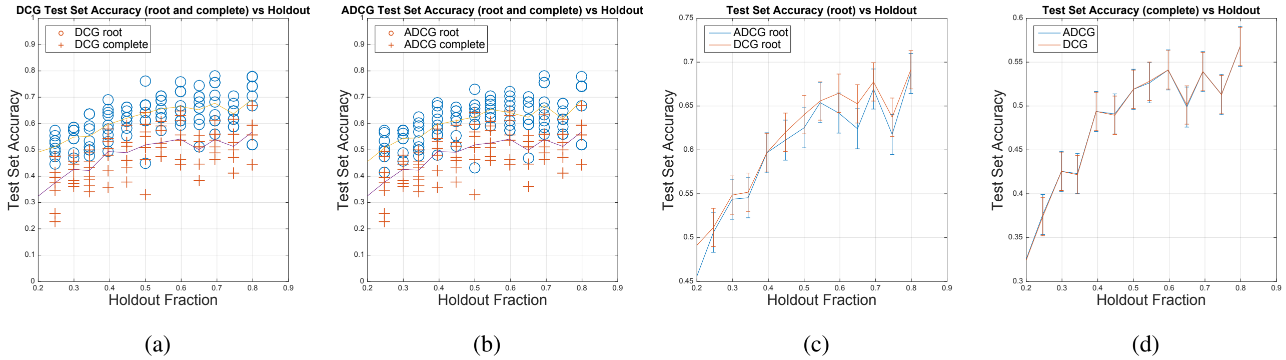
Instead of exhaustively searching over the entire abstract search space, the procedure uses the expressed concrete groundings to selectively instantiate a restricted space of probable abstract symbols that is then explored for solutions. The estimated object groundings are used to estimate a reduced space of probable abstractions: $\widehat{\eta(i)} = \{(\sigma_i, \widehat{O_j}) | \sigma_i \in \Sigma, \widehat{O_j} \subseteq O\}$. Note that each abstract factor is conditioned on the expressed concrete groundings as well as groundings from child phrases. Instead of exhaustively searching over the entire abstract search space, the procedure uses the expressed concrete groundings to selectively instantiate a restricted space of probable abstract symbols that is then explored for solutions. The approximate abstract search space $\Gamma^{\widehat{\mathcal{A}(i)}}$ is not fixed, but varies dynamically per phrase λ_i and is determined based on the estimated true concrete groundings. The induced abstract factors constitute the following approximate joint distribution:

$$\begin{aligned}\underset{\phi^{C}_{ij},\phi^{\mathcal{A}}_{ik}\in\mathbf{\Phi}}{\arg\max}\Bigg\{ \prod_{i=1}^{|\mathcal{N}|} \Bigg(\prod_{j=1}^{|C|} \Psi(\phi^{C}_{ij}, \gamma^{C}_{ij}, \lambda_i, \{\mathbf{\Phi}^{C}_{c_i} \cup \mathbf{\Phi}^{\widehat{\mathcal{A}}}_{c_i}\}, \Upsilon) \\ \prod_{k=1}^{|\widehat{\mathcal{A}(i)}|} \Psi(\phi^{\widehat{\mathcal{A}(i)}}_{ik}, \gamma^{\widehat{\mathcal{A}(i)}}_{ik}, \lambda_i, \{\mathbf{\Phi}^{C}_{i} \cup \mathbf{\Phi}^{C}_{c_i} \cup \mathbf{\Phi}^{\widehat{\mathcal{A}}}_{c_i}\}, \Upsilon)\Bigg)\Bigg\}\end{aligned} \tag{5}$$

4 Evaluation

The proposed model was evaluated using a corpus generated from a user study.

The corpus consisted of natural language descriptions paired with simulated scenes demonstrating a Baxter robot carrying out manipulation tasks with varying blocks arrangements, Figure 3. The language descriptions were provided by human subjects via the Amazon Mechanical Turk platform. The data set consisted of 135 language descriptions, each paired with spatial context arising from 21 randomized workspaces resulting in a total of 1672 annotated phrases. The input instructions were tagged with part-of-speech labels from the Penn Tagset [Marcus *et al.*, 1993] and parsed using the Cocke-Kasami-Younger (CKY) algorithm [Younger, 1967]. The proposed ADCG model was compared against the DCG [Howard *et al.*, 2014b] model as a baseline. For a fair



(a) (b) (c) (d)

Figure 4: Test set accuracy (y-axis) vs. holdout fraction (x-axis) using the root-phrase metric (root phrase correctly grounded) and complete-tree metric (all phrases correctly grounded). (a) The DCG model accuracy for both root-phrase and complete-tree accuracy metrics. (b) The ADCG model accuracy using both root-phrase and complete-tree accuracy metrics. (c) The average root-phrase accuracy for both models. (d) The average complete-tree accuracy for both models. Note the scale on the y-axis. The accuracy of the proposed ADCG model closely followed the DCG baseline. The holdout fraction varied between 0.2 to 0.8 in increments of 0.05 with 9 runs for each fraction.

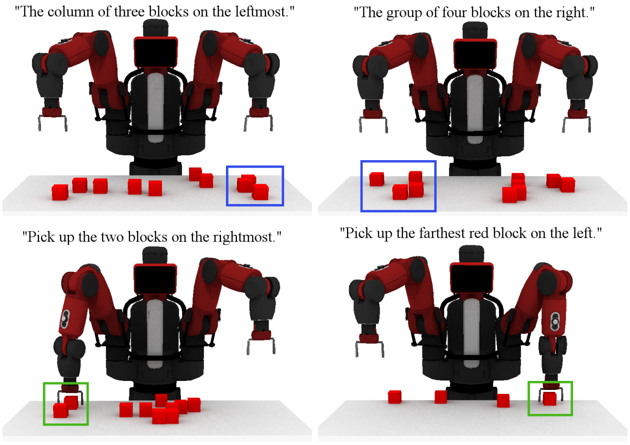

"The column of three blocks on the leftmost."
"The group of four blocks on the right."
"Pick up the two blocks on the rightmost."
"Pick up the farthest red block on the left."

Figure 3: Examples of natural language descriptions in our aligned corpora that were collected using Amazon Mechanical Turk and our simulation environment.

comparison, the DCG search space was expanded to include all possible abstract concepts (the power set of concrete constituents) as independent without being hierarchically linked with concrete groundings. The log-linear model training step, feature sets and training remained the same for both models.

Figure 4 compares the grounding accuracy for instructions in the corpus for the proposed ADCG model and the baseline DCG model. Training was carried out using randomly sampled subsets and increasing the holdout fraction from 0.2 to 0.8 in increments of 0.05 with 9 runs for each fraction. Maximum probable groundings (above a 0.75 threshold) were determined for each phrase in the parsed instruction. The ADCG accuracy closely followed the DCG baseline. The model correctly grounded instructions like, "the second farthest block to the left in the row of blocks in front of the robot". References to abstract concepts in the context of spatial and numeric information like "horizontal line of four blocks", "farthest three blocks", "row of blocks in front", "group of eight blocks", "nearest two blocks" etc. are correctly grounded. References to constituent elements like "middle", "second farthest", "nearest" are also correctly inferred by the system.

Table 1: Average Inference Runtime for Corpus

			Runtime (seconds)	
Objects	Instructions	Worlds	DCG	Proposed ADCG Model
4	4	1	0.14 ± 0.003	$0.007 \pm 2.3 \times 10^{-4}$
5	45	9	0.21 ± 0.009	$0.009 \pm 5.7 \times 10^{-4}$
7	62	5	0.47 ± 0.033	$0.010 \pm 7.9 \times 10^{-4}$
10	10	5	2.96 ± 0.177	$0.010 \pm 1.0 \times 10^{-4}$
12	13	1	14.25 ± 0.510	$0.011 \pm 7.2 \times 10^{-4}$
Total	134	21	1.89 ± 4.12	$0.062 \pm 1.0 \times 10^{-3}$

Table 1 presents the total average inference runtime normalized by the number of phrases per instruction for the corpus. The ADCG model has significantly lower average runtime than the DCG baseline. The runtime gain for the ADCG model is more pronounced with greater scene complexity. The runtime for grounding abstract concepts is determined by the total size of the grounding space searched for solutions. The ADCG model is able to determine a reduced space of probable hypotheses and hence searches an approximate reduced space compared to the DCG baseline model that exhaustively searches for solutions. This approximation leads to a significant efficiency gain with minimal loss in accuracy, as demonstrated in Figure 4. We believe that the technique of ordering factor computations and exploiting the conditional structure of hierarchical concepts linked with concrete concepts to estimate a smaller space of probable concepts may have applicability for search in other domains where similar structure exists.

Acknowledgements

This work was supported in part by the Robotics Consortium of the U.S Army Research Laboratory under the Collaborative Technology Alliance Program and by the National Science Foundation under Grant No.1427547.

References

[Boularias *et al.*, 2015] Abdeslam Boularias, Felix Duvallet, Jean Oh, and Anthony Stentz. Grounding spatial relations for outdoor robot navigation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1976–1982, 2015.

[Howard *et al.*, 2014a] Thomas M Howard, Istvan Chung, Oron Propp, Matthew R Walter, and Nicholas Roy. Efficient natural language interfaces for assistive robots. In *IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS) Work. on Rehabilitation and Assistive Robotics*, 2014.

[Howard *et al.*, 2014b] Thomas M Howard, Stefanie Tellex, and Nicholas Roy. A natural language planner interface for mobile manipulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6652–6659. IEEE, 2014.

[Marcus *et al.*, 1993] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[Matuszek *et al.*, 2012] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. In *Proceedings of the 13th International Symposium on Experimental Robotics (ISER)*, June 2012.

[Oh *et al.*, 2015] Jean Oh, Arne Suppé, Felix Duvallet, Abdeslam Boularias, Jerry Vinokurov, Oscar Romero, Christian Lebiere, and Robert Dean. Toward mobile robots reasoning like humans. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1371–1379, 2015.

[Paul *et al.*, 2016] Rohan Paul, Jacob Arkin, Nicholas Roy, and Thomas M. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[Tellex *et al.*, 2011] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth J Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2011.

[Younger, 1967] Daniel H Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2):189–208, 1967.