Efficient Grounding of Abstract Spatial Concepts for Natural Language Interaction with Robot Manipulators

Rohan Paul[1], Jacob Arkin[2], Nicholas Roy[1] and Thomas M. Howard[2]

***Abstract*—Our goal is to develop models that allow a robot to understand natural language instructions in the context of its world representation. Contemporary models learn possible correspondences between parsed instructions and candidate groundings that include objects, regions and motion constraints. However, these models cannot reason about abstract concepts expressed in an instruction like, "pick up the middle block in the row of five blocks". In this work, we introduce a probabilistic model that incorporates an expressive space of abstract spatial concepts as well as notions of cardinality and ordinality. The graph is structured according to the parse structure of language and introduces a factorisation over abstract concepts correlated with concrete constituents. Inference in the model is posed as an approximate search procedure that leverages partitioning of the joint in terms of concrete and abstract factors. The algorithm first estimates a set of probable concrete constituents that constrains the search procedure to a reduced space of abstract concepts, pruning away improbable portions of the exponentially-large search space. 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.**

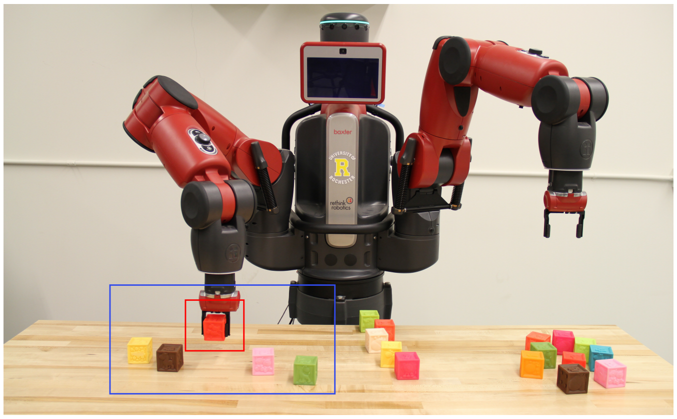


Fig. 1: Robot following the instruction, "pick up the middle block in the row of five blocks on the right". The grounding for an aggregative concept ("rows") is abstract and linked with the expression of constituent concrete groundings ("blocks"). The space of abstract concepts is exponentially-large 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.

I. Introduction

Advances in autonomy have allowed 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. Hence, a robot operating alongside humans must possess the ability to infer intent, task constraints and workspace knowledge conveyed in natural language instructions.

The problem of assigning meaning or "grounding" natural language instructions is challenging due to the complexity of concepts expressed via human language and the diversity of workspaces the robot may be operating in. Significant progress has been made in the development of probabilistic models [1]–[5] that associate natural language instructions (noun, preposition and verb phrases) with semantic entities in the world (objects, regions and actions). State of the art models pose the grounding problem as inference on a probabilistic graph and can successfully infer groundings and hence plans from instructions such as "put the pallet on the the truck" or "pick up the block on the table".

The expressiveness of contemporary models is restricted to concrete entities that partition the robot's state-action space. A key limitation is the lack of a notion of abstractions, pervasive in human language. Consider the following instructions, "pick up the second block from the row of blocks", "keep visible the circle of vehicles", "exit through the third door" or "clear the group of blocks in front" etc.

In this work, we present the Adaptive Distributed Correspondence Graph (ADCG) model that incorporates a generalised space of groundings encompassing abstract concepts. Abstractions are expressed as hierarchical groundings that are probabilistically linked to the expression of concrete constituents. The model also incorporates notions of cardinality ("one", "two", "three" etc.), ordinality ("fifth", "sixth", "seventh" etc.) or spatial references ("farthest", "nearest", "left-most" etc.) for resolving constituents within abstract concepts. The joint distribution factorises both according to the parse structure of the input language and hierarchically over abstract and concrete factors. Training is accomplished in a data-driven manner using predictors based on spatial, language and contextual grounding cues.

A key technical challenge is efficient inference in the proposed model since the domain of abstract groundings grows exponentially in the number of concrete groundings. We present an approximate search procedure that orders factor evaluations such that likely concrete hypothesis are estimated first. Conditioned on expressed concrete groundings

[1] Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Cambridge, MA 02139, USA {rohanp,nickroy}@csail.mit.edu

[2] Robotics and Artificial Intelligence Laboratory (RAIL), Electrical and Computer Engineering, University of Rochester, Rochester NY 14623, USA {j.arkin, thomas.howard}@rochester.edu

a reduced space of probable abstract concepts is determined, pruning away improbable portions of the search space. The abstract grounding variables in the model render the individual concrete groundings conditionally independent of the other groundings in the model. This permits an adaptive instantiation of a reduced probable search space of abstract symbols conditioned on likely concrete constituents, making efficient approximate inference possible.

We evaluate the model using a corpus consisting of simulated scenes from a manipulation domain paired with language descriptions obtained from a user study. Results demonstrate effectiveness in learning abstract concepts and successfully grounding commands possessing nested and composed abstract references. Empirical evaluation reveals significantly lower runtime for the model with equivalent accuracy compared to the state of the art baseline. The system was also deployed on a Baxter Research Robot.

II. Grounding Natural Language Instructions

The task of grounding natural language instructions involves assigning "meaning" to input phrases in the context of the robot's world model and action space. Let (Υ) denote the physical workspace of the robot that aggregates metric and semantic information about constituent objects. The set of groundings (Γ) consists of a set of symbols that correspond to semantic notions such as objects, locations, regions, paths or actions the robot can take. It is common to associate noun phrases with objects, prepositional phrases with regions and verb phrases with a set of actions or motion constraints. For example, the set of groundings for the instruction "pick up the block on the table" include objects for phrases "the block" and "the table", "on" is interpreted as a region above the "table" object and "pick up" determines the intended grasp action to be executed by the robot. The input 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 the input instruction:

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

Contemporary techniques like Generalised Grounding Graphs (G3) [2] model Equation 1 as inference on a factor graph structured according to the parse structure of the input instruction. Binary correspondence variables (ϕ_i) express likely association between a phrase (λ_i) and candidate grounding (γ_i). The model assumes conditional independence between groundings for a phrase given child groundings $(\mathbf{\Gamma}_{c_i})$ permitting factorisation across phrases. Probable groundings are determined by maximising the conditional likelihood of true correspondences while sampling over the unknown candidate groundings:

$$\underset{\gamma_1 \cdots \gamma_n \in \mathbf{\Gamma}}{\arg\max} \; \prod_{i=1}^{|\mathcal{N}|} p(\phi_i = True | \gamma_i, \lambda_i, \mathbf{\Gamma}_{c_i}, \Upsilon). \tag{2}$$

The Distributed Correspondence Graph (DCG) [6] model, discretises the continuous space of groundings as regions and motion constraints, ameliorating the need to sample trajectories or locations. Correspondence variables (ϕ_{ij}) relate the i^{th} phrase (λ_i) with the $\jmath^{th}$ grounding constituent (γ_{ij}). The model distributes grounding constituents as conditionally independent factors and structures inference as a search over unknown correspondences:

$$\underset{\phi_{ij} \in \mathbf{\Phi}}{\arg\max} \; \prod_{i=1}^{|\mathcal{N}|} \prod_{j=1}^{|\mathcal{C}|} p(\phi_{ij} | \gamma_{ij}, \lambda_i, \mathbf{\Phi}_{c_i}, \Upsilon). \tag{3}$$

The factor graph representation appears in Figure 2(a). Contemporary approaches only model concrete entities in the world model and hence implicitly assume a flat non-hierarchical space of groundings. This representation is insufficient for grounding spatial abstractions whose meaning is interpreted hierarchically in terms of concrete constituent entities. In this work, we develop a probabilistic model that can reason about abstractions. We proceed by formulating a generalised space of groundings in the next section followed by the graphical model in Section IV.

III. Generalised Space of Groundings

In this section, we develop a symbolic representation of the robot's state-action space including notions of spatial abstractions. The symbol set represents the space of groundings (Γ) in which the input language instruction $(\mathbf{\Lambda})$ is interpreted.

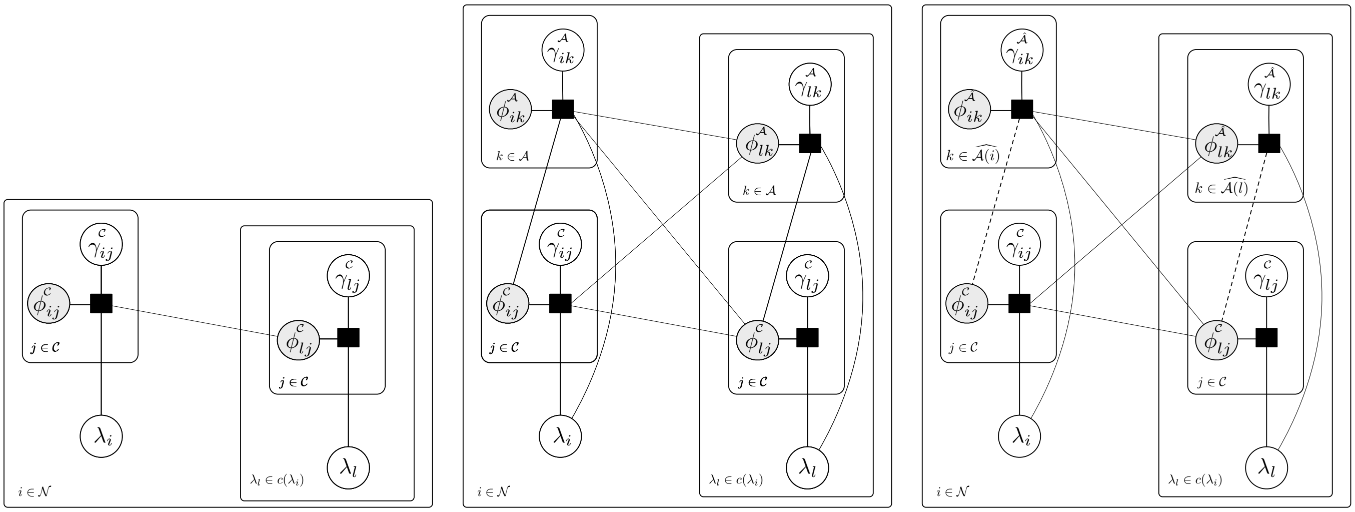
A. Concrete Groundings

A set of concrete symbols model constituents of the robot's state-action space. The robot's workspace is represented as the set of objects $(\mathcal{O})$ within perceptual view, each possessing geometric, appearance and pose information. The set $(\mathcal{L})$ denotes semantic labels ("red block", "robot hand", "table" etc.) typically classification outputs from a perception system. We assume that the robot is capable of executing a set of manipulation actions ("pick", "place", "clear" etc.) forming the set (Δ) parameterised by the objects under consideration. Next, we introduce the following sets of grounding symbols that range over objects, labels and actions: $\Gamma^{\mathcal{O}} = \{\gamma_{o_i} | o_j \in \mathcal{O}\}$, $\Gamma^{\mathcal{L}} = \{\gamma_{l_i} | l_i \in \mathcal{L}\}$ and $\Gamma^{\Delta} = \{\gamma_{\alpha_i} | \alpha_i \in \Delta\}$.

Further, we assume the presence of spatial references ("left", "center", "behind" etc.) denoted by the set $(\mathcal{S})$. We also incorporate notions of cardinality ("two", "three", "four" etc.) and ordinality ("fifth", "sixth", "seventh" etc.) represented as the sets (χ) and $(\mathcal{H})$ respectively. We associate the concrete entities defined above with the set of grounding symbols defined as: $\Gamma^{\mathcal{S}} = \{s_i | s_i \in \mathcal{S}\}$, $\Gamma^{\chi} = \{\kappa_i | \kappa_i \in \chi\}$ and $\Gamma^{\mathcal{H}} = \{h_i | h_i \in \mathcal{H}\}$.

The set of concrete groundings is sufficient capture groundings for phrases such as "four blocks", "fifth item on the left", "pick up the red block" etc. The union of symbol set cumulatively forms the space of concrete symbols $(\Gamma^{\mathcal{C}})$ as:

$$\Gamma^{\mathcal{C}} = \{\Gamma^{\mathcal{O}} \cup \Gamma^{\mathcal{L}} \cup \Gamma^{\mathcal{H}} \cup \Gamma^{\chi} \cup \Gamma^{\mathcal{S}} \cup \Gamma^{\Delta}\}. \tag{4}$$


$\gamma^{\mathcal{A}}_{ik}$
$\phi^{\mathcal{A}}_{ik}$
$k \in \mathcal{A}$
$k \in \widehat{\mathcal{A}(i)}$
$\gamma^{\mathcal{A}}_{lk}$
$\phi^{\mathcal{A}}_{lk}$
$k \in \widehat{\mathcal{A}(l)}$
$\gamma^{\mathcal{C}}_{ij}$
$\phi^{\mathcal{C}}_{ij}$
$j \in \mathcal{C}$
$\gamma^{\mathcal{C}}_{lj}$
$\phi^{\mathcal{C}}_{lj}$
λ_i
λ_l
$\lambda_l \in c(\lambda_i)$
$i \in \mathcal{N}$

(a) DCG Model

(b) Proposed ADCG Model

(c) ADCG Model Approximation

Fig. 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) 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. (c) The approximate ADCG model used in inference. A reduced space of abstract factors ($\widehat{\mathcal{A}}$) is selectively instantiated based on the expressed distribution for concrete groundings (dashed line) given child context.

B. Abstract Groundings

Abstractions are introduced as hierarchical symbols expressed as an aggregation of constitutent concrete symbols.

This includes the notion spatial aggregations or containers ("rows", "columns", "groups", "towers" etc.) conveyed in instructions like "the column of red blocks". The set of containers (η) is formed as a subset of objects ($\mathcal{O}_j \subseteq \mathcal{O}$) possessing a common spatial characteristic ("linearity", "circularity", "directivity" etc.) denoted by the set (Σ):

$$\eta = \{(\sigma_i, \mathcal{O}_j) | \sigma_i \in \Sigma, \mathcal{O}_j \subseteq \mathcal{O}\}. \quad (5)$$

The induced space of grounding symbols for abstract containers is expressed as $\Gamma^{\eta} = \{\gamma_{(\sigma_j, \mathcal{O}_k)} | (\sigma_j, \mathcal{O}_k) \in \eta\}$. Note that the number of possible containers is $(|\Sigma| \times |\mathcal{P}(\mathcal{O})|)$, where $\mathcal{P}$ denotes the power set. Hence, the symbol space of containers and associated regions is exponential $\mathcal{O}(2^{N_{\mathcal{O}}})$ in the number of objects populating the world model.

Another category of hierarchical symbols model attributes associated with objects or containers. Consider the phrase, "the middle block" in the instruction depicted in Figure 1. The phrase grounding relates an object label ("block") with an abstract spatial reference ("middle"), that can be considered an associated attribute. Such symbolic aggregations are abstract, i.e., cannot be physically grounded to an object until it is composed with a phrase like, "in the row" that provide context. Additionally, abstract attributes can include index information, for example, "the fifth block from the left". We define abstract object attributes $\Pi_{\mathcal{O}}$ as:

$$\Pi_{\mathcal{O}} = \{(l_i, \kappa_j, h_k) | l_i \in \mathcal{L}, \kappa_j \in \chi, h_k \in \mathcal{H}\}. \quad (6)$$

Attributes can also be expressed in the context of containers, for example, "the seven blocks in front" that associate index ("first") and cardinal ("seven") symbol with a collection of objects ("blocks") till contextual symbols are grounded to provide reference. The notion of abstract container attributes is defined as:

$$\Pi_{\eta} = \{(l_i, h_j, s_k) | l_i \in \mathcal{L}, h_j \in \mathcal{H}, s_k \in \mathcal{S}\}. \quad (7)$$

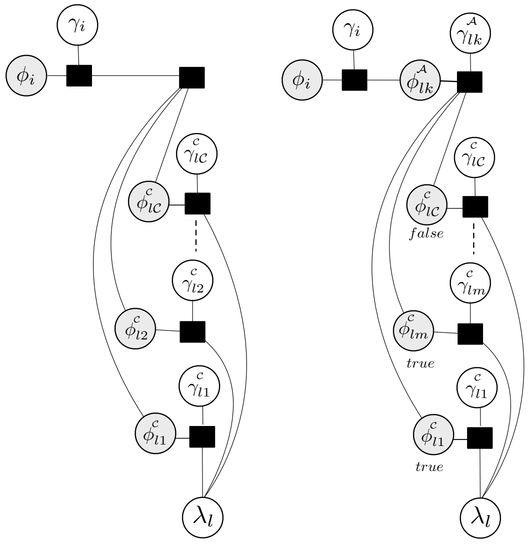
The corresponding grounding sets, $\Gamma^{\Pi_{\mathcal{O}}}$ and $\Gamma^{\Pi_{\eta}}$ are included as: $\Gamma^{\Pi_{\mathcal{O}}} = \{\gamma_{(l_i, \kappa_j, h_k)} | (l_i, \kappa_j, h_k) \in \Pi_{\mathcal{O}}\}$ and $\Gamma^{\Pi_{\eta}} = \{\gamma_{(l_i, h_j, s_k)} | (l_i, h_j, s_k) \in \Pi_{\eta}\}$. Further, the set of spatial relations associated with containers ("center of a ring" or "right-side of the column") are included as: $\Gamma^{\mathcal{R}_{\eta}} = \{\gamma^{s_i}_{(\sigma_j, \mathcal{O}_k)} | s_i \in \mathcal{S}, (\sigma_j, \mathcal{O}_k) \in \eta\}$. The symbolic representation presented above forms the space of abstract groundings ($\Gamma^{\mathcal{C}}$):

$$\Gamma^{\mathcal{A}} = \{\Gamma^{\eta} \cup \Gamma^{\Pi_{\mathcal{O}}} \cup \Gamma^{\Pi_{\eta}} \cup \Gamma^{\mathcal{R}_{\eta}}\}. \quad (8)$$

C. Space of Groundings

The concrete and abstract symbol spaces collectively form the generalised space of groundings ($\Gamma = \Gamma^{\mathcal{C}} \cup \Gamma^{\mathcal{A}}$). The size of the concrete grounding space grows linearly as $\mathcal{O}(N_{\mathcal{O}})$. However, the space of abstract groundings is exponentially large in the number of concrete symbols $\mathcal{O}(2^{|\mathcal{C}|})$. This is determined primarily by the number of possible abstract containers $\mathcal{O}(2^{N_{\mathcal{O}}})$ growing exponentially with $N_{\mathcal{O}}$.

This poses a significant challenge as the robot manipulator may be operating in complex environments possessing a large number of objects with diverse spatial layouts. Even in a simplistic block world setup with 20 objects, Figure 1, results in an abstract search space that includes 17.3 million symbols.


γ_i
ϕ_i
$\overset{c}{\gamma}_{lC}$
$\overset{c}{\phi}_{lC}$
$\overset{c}{\gamma}_{l2}$
$\overset{c}{\phi}_{l2}$
$\overset{c}{\gamma}_{l1}$
$\overset{c}{\phi}_{l1}$
λ_l
γ_i
$\overset{\mathcal{A}}{\gamma}_{lk}$
ϕ_i
$\overset{\mathcal{A}}{\phi}_{lk}$
$\overset{c}{\gamma}_{lC}$
$\overset{c}{\phi}_{lC}$
false
$\overset{c}{\gamma}_{lm}$
$\overset{c}{\phi}_{lm}$
true
$\overset{c}{\gamma}_{l1}$
$\overset{c}{\phi}_{l1}$
true
λ_l

(a) (b)

Fig. 3: Approaches for modelling abstract concepts. (a) Introduction of a joint factor between concrete correspondences ($\phi^{\mathcal{C}}$) or (b) Inclusion of abstract correspondence variables ($\phi^{\mathcal{A}}$), each associated with a unique subset of true concrete correspondences. An abstract variable ($\phi^{\mathcal{A}}_{ik}$) acts as an indicator for an expressed concrete aggregation $\{\phi^{\mathcal{C}}_{l1}$ $\ldots\phi^{\mathcal{C}}_{lm}\}$. Note that given the knowledge of an abstract grounding ($\phi^{\mathcal{A}}_{lk}$), a parent correspondence (ϕ_i) is independent of the set of concrete correspondences $\{\phi^{\mathcal{C}}_{l1},\ \ldots\phi^{\mathcal{C}}_{lC}\}$. This decorrelation allows significant reduction in factor training and inference complexity.

IV. Inference over the Probabilistic Model

This section presents the Adaptive Distributed Correspondence Graph (ADCG) model that incorporates the expressive symbol space introduced previously. This is followed by a description of the log-linear model and features that enable learning of abstract concepts from spatial and language cues. We then address the crucial issue of efficient inference in the exponentially-large space of abstract symbols and describe the generation of a reduced adaptive abstract search space.

A. Variables

Estimating the likely correspondences (Φ) between the input natural language instruction ($\mathbf{\Lambda}$) and the generalised space of groundings ($\Gamma^{\mathcal{C}} \cup \Gamma^{\mathcal{A}}$) is posed as inference on a factor graph. Following the DCG formulation (Section II) concrete correspondence variables ($\overset{\mathcal{C}}{\phi}_{ij}$) are introduced that relate a phrase (λ_i) with concrete groundings ($\overset{\mathcal{C}}{\gamma}_{ij}$).

Next, we introduce abstract concepts as hierarchical symbols correlated with concrete groundings. All possible aggregations of concrete groundings constitute the space of abstract concepts. Figure 3(a) illustrates an approach for modelling this relationship by introducing a shared factor between all concrete correspondences. Although, the factor captures the joint association of concrete groundings it loses distinguishability in resolving multiple aggregative groundings. For example, a phrase like, “the first two blocks in the row” and “the third farthest blocks in the column” would ground to a set of five expressed objects without expressing the semantics of being associated with a container. Further, grounding a hierarchically linked phrase such as, “the middle block in the row” would necessitate reasoning jointly about the full set of objects as well as references to a subset within the collections.

This motivates the inclusion of explicit abstract grounding variables ($\overset{\mathcal{A}}{\gamma}_{ik}$) and correspondences ($\overset{\mathcal{A}}{\phi}_{ik}$), Figure 3(b). Each variable expresses a unique subset of concrete correspondences. Although, this representation has the same expressiveness as before, it has the advantage that the abstract correspondences act as indicator functions distinguishing each expressed aggregation. Further, conditioning on the abstract variable, a hierarchically linked grounding, like “the middle block” is de-correlated with the joint distribution over concrete constituents. This reduces the complexity of factor training at the expense of introducing more grounding factors (which we address this in the next section). The set of abstract correspondence variables form a Markov boundary for the concrete correspondences.

B. Factorisation

Factors in the model are partitioned as concrete and abstract. Each factor relates unknown correspondences, an input phrase and a probable grounding as well as the set of correspondences from child phrases. Let the variable sets ($\mathbf{\Phi}^{\mathcal{C}}_{c_i}$) and ($\mathbf{\Phi}^{\mathcal{A}}_{c_i}$) denote concrete and abstract correspondences for all child phrases for the parent phrase (λ_i). Conditioning the concrete factor on ($\mathbf{\Phi}^{\mathcal{C}}_{c_i}$) allows grounding of concrete objects that possess pairwise relationships, e.g. “near the red block and blue block” whereas the set ($\mathbf{\Phi}^{\mathcal{A}}_{c_i}$), provides context for grounding a constituent element referenced as “the farthest block in the group”.

An abstract factor links the input phrase with correspondences related to abstract groundings. Let the variable set ($\mathbf{\Phi}^{\mathcal{C}}_{i}$) denote the concrete correspondences for the current phrase. The set cumulatively represents concrete groundings inferred from child context for the current phrase. Each abstract factor is shared with ($\mathbf{\Phi}^{\mathcal{C}}_{i}$), expressing the correlation of a hierarchical abstract grounding with the set of concrete groundings, e.g. “column of blocks on the right”. The links connecting the abstract factors between parent and child phrases facilitate reasoning over relationships between abstract groundings,“between the row and the column”.

The above construction is followed for each parent and child phrase resulting in the full grounding graph structured according to the parse tree of the input instruction. The model distributes conditionally independent grounding elements across multiple factors for input phrases. Concrete correspondences phrases are considered independent given child groundings and conditioned on the concrete groundings for a phrase. The abstract correspondences are independent given the concrete groundings and child groundings. The grounding elements are known and the model searches over unknown correspondences. Figure 2(b) presents the factor graph. The distribution joint distribution is modelled as a product of factor potentials (Ψ) given as:

$$\underset{\phi^{\mathcal{C}}_{ij},\phi^{\mathcal{A}}_{ik}\in\mathbf{\Phi}}{\arg\max}\left\{\prod_{i=1}^{|\mathcal{N}|}\left(\prod_{j=1}^{|\mathcal{C}|}\Psi(\phi^{\mathcal{C}}_{ij},\gamma^{\mathcal{C}}_{ij},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\mathcal{A}}_{c_i}\},\Upsilon)\right.\right.$$

$$\left.\left.\prod_{k=1}^{|\mathcal{A}|}\Psi(\phi^{\mathcal{A}}_{ik},\gamma^{\mathcal{A}}_{ik},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{i}\cup\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\mathcal{A}}_{c_i}\},\Upsilon)\right)\right\}. \tag{9}$$

C. Factor Potentials

The factors in the model are are evaluated using a log-linear model. Each factor potential (Ψ) is expressed as a linear combination of binary feature functions (f_l) predicting association between the constituent variables. The feature weights (μ_l) are learned using a labeled corpus.

$$\psi(\phi^{\mathcal{A}}_{ik},\gamma^{\mathcal{A}}_{ik},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{i}\cup\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\mathcal{A}}_{c_i}\},\Upsilon)= \frac{\exp^{\sum_l \mu_l f_l(\phi^{\mathcal{A}}_{ik},\gamma^{\mathcal{A}}_{ik},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{i}\cup\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\mathcal{A}}_{c_i}\},\Upsilon)}}{\sum_q \exp^{\sum_l \mu_l f_l(\phi^{\mathcal{A}}_{q},\gamma^{\mathcal{A}}_{ik},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{i}\cup\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\mathcal{A}}_{c_i}\},\mathbf{\Gamma}^{\boldsymbol{\mathcal{A}}}_{\bar{\boldsymbol{c}}_{\boldsymbol{ik}}}\},\Upsilon)}} \tag{10}$$

A set of geometric features are introduced that capture spatial characteristics of object aggregations in the workspace. These included shape likelihood (inlier fraction), model fit (statistics for deviation from a linear fit), element coherence (uniformity of object types in the aggregation), separation (inter-element distances), ordering (relative positioning among constituent elements), orientation (along principal axes) and model separation (distance to nearest non-member elements). Features are computed using linear and square exponential kernels. The second set of features capture lexical cues and part-of-speech information determined from parsing. Additional features capture the association of abstract groundings in the context of expressed concrete child groundings propagated from nested child phrases in the tree-structured graph. The continuous features are scale-normalised and discretised using uniform sampling to output binary values. A total of 50,820 binary features were used in the log-linear model. Feature weights are learnt by maximising the training set likelihood using a stochastic gradient descent routine L-BFGS [7].

D. 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. This necessitates the use of approximate techniques. In order to search efficiently, we leverage a partitioning of the joint distribution between concrete and abstract factors, Equation 9. 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. For example, the set of possible object groundings are obtained as:

$$\widehat{\mathcal{O}(i)}=\{\widehat{o_j}|\gamma^{\mathcal{C}}_{ij}=\widehat{o_j},p(\phi^{\mathcal{C}}_{ij}|\gamma^{\mathcal{C}}_{ij},\lambda_i,\mathbf{\Phi}_{\boldsymbol{c_i}})\geq p_T,j\in\mathcal{C}\}. \tag{11}$$

The next step is to search over the space of abstract factors. 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. For example, the likely object groundings deterministically generate a set of container hypothesis, generating a reduced search space for evaluation:

$$\widehat{\eta(i)}=\{(\sigma_i,\widehat{\mathcal{O}_j})|\sigma_i\in\Sigma,\widehat{\mathcal{O}_j}\subseteq\mathcal{O}\}. \tag{12}$$

Similarly, the space of probable abstract object and container attribute symbols can also be constrained based on concrete groundings. The approximate abstract search space $\Gamma^{\widehat{\mathcal{A}(i)}}$ is not fixed, but varies dynamically per phrase λ_i, constructed based on the probable constituents of the distribution over concrete symbols:

$$\Gamma^{\widehat{\mathcal{A}(i)}}=\{\Gamma^{\widehat{\eta(i)}}\cup\Gamma^{\widehat{\Pi_{\mathcal{O}}(i)}}\cup\Gamma^{\widehat{\Pi_{\eta}(i)}}\cup\Gamma^{\widehat{\mathcal{R}_{\eta}(i)}}\} \tag{13}$$

The solutions for both concrete and abstract correspondences are then combined and passed as child groundings up the search tree to parent nodes. The search process is implemented in a bottom-up manner, that starts grounding the leaf phrases upwards. The procedure continues till the root phrase is grounded, forming the grounding solution for the entire instruction. Figure 2(c) presents the factor graph for the approximate model used for inference. The induced abstract factors constitute the following approximate joint distribution for the model:

$$\underset{\phi^{\mathcal{C}}_{ij},\phi^{\mathcal{A}}_{ik}\in\mathbf{\Phi}}{\arg\max}\left\{\prod_{i=1}^{|\mathcal{N}|}\left(\prod_{j=1}^{|\mathcal{C}|}\Psi(\phi^{\mathcal{C}}_{ij},\gamma^{\mathcal{C}}_{ij},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\widehat{\mathcal{A}}}_{c_i}\},\Upsilon)\right.\right.$$

$$\left.\left.\prod_{k=1}^{|\widehat{\mathcal{A}(i)}|}\Psi(\phi^{\widehat{\mathcal{A}(i)}}_{ik},\gamma^{\widehat{\mathcal{A}(i)}}_{ik},\lambda_i,\{\mathbf{\Phi}^{\mathcal{C}}_{i}\cup\mathbf{\Phi}^{\mathcal{C}}_{c_i}\cup\mathbf{\Phi}^{\widehat{\mathcal{A}}}_{c_i}\},\Upsilon)\right)\right\} \tag{14}$$

Note that the knowledge of abstract correspondences makes the individual concrete correspondences conditionally independent of the rest of the model. This allows reasoning about concrete constituents first for proposing an approximate space of abstract concepts, independently of the remaining model. Although, in the worst case, the entire abstract space may need to be searched, in practice, the size of the adaptive search space is significantly smaller, leading to a significant reduction in inference time. Note that search is approximate. Firstly, the set of likely candidate groundings passed as child groundings is bounded (beam-width parameter). Further, the constrained search space of abstract symbols may directly exclude hypotheses for which concrete groundings are uncertain. However, we establish empirically in the next section that the approximation leads to significant efficiency gains with a minimal loss in accuracy.

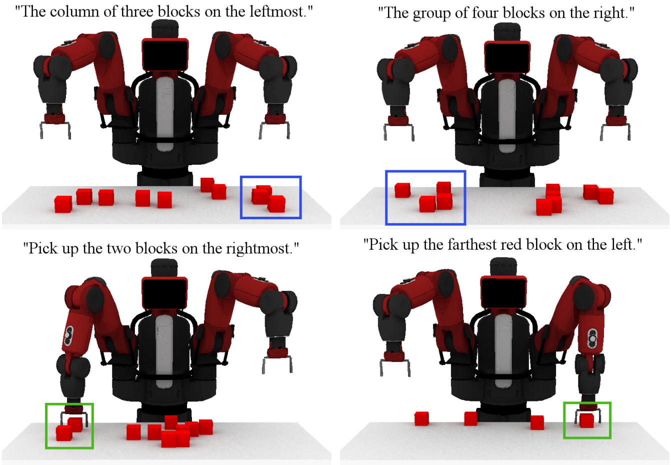

"The column of three blocks on the leftmost."
"The group of four blocks on the right."
"Pick up the two blocks on the rightmost."
"Pick up the farthest red block on the left."

Fig. 4: Examples of natural language descriptions in our aligned corpora that were collected using Amazon Mechanical Turk and our simulation environment.

V. Evaluation

The proposed model was trained and evaluated using a data corpus from a user study. This section outlines the data collection and experimental setup.

A. Corpus

The data corpus consisted of natural language descriptions paired with simulated scenes showing a Baxter robot carrying out manipulation tasks with varying blocks arrangements, Figure 4. The relative positions of objects and the robot's end effector were randomised by applying random force vectors and simulating the physical effects for a short duration (2.5 sec) until stable configurations were obtained. A set of 128 images were generated and overlaid with bounding boxes indicating all visible spatial groupings and constituent objects within the collection. Further, a data set consisting of 8 short video sequences (3-7 seconds) was collected showing the robot executing pick, place and clearing tasks in a randomised workspaces.

The language descriptions were obtained from human subjects via the Amazon Mechanical Turk platform.Subjects were presented with the image sequences and requested to describe the marked objects in the context of the spatial neighbourhood and the robot's end effector position. For the video data set, subjects were asked to provide a natural language command that would generate the observed behaviour of the robot. Subjects were presented the data only once and on average 10 language descriptions were obtained for each image or video. The final corpus included a total of 135 language descriptions, each paired with spatial context arising from 21 randomised workspaces resulting in a total of 1672 individual phrases. The input instructions were tagged with part-of-speech labels from the Penn Tagset [8] and parsed using the Cocke-Kasami-Younger (CKY) algorithm [9]. The parsed instructions were annotated with ground truth grounding sets. All incorrect combinations provided negative training data.

B. Experimental Setup and Baseline

The proposed model was compared against the DCG [1] model as a baseline. For a fair comparison, the search space was expanded to include all possible abstract concepts (the power set of concrete constituents). Independent factors were introduced linking phrases with concrete and abstract groundings respectively. Note that this factorisation makes the abstract groundings independent of base groundings and follows directly the non-hierarchical inclusion of grounding symbols in the DCG model. The search space was fixed and searched exhaustively for all phrases. The log-linear model training step, feature sets and training remained the same for both models. All tests were run on a Quad core Intel Core i5 processor with 16 GB of RAM running Mac OS X 10.11 (x86-64) architecture.

VI. Results

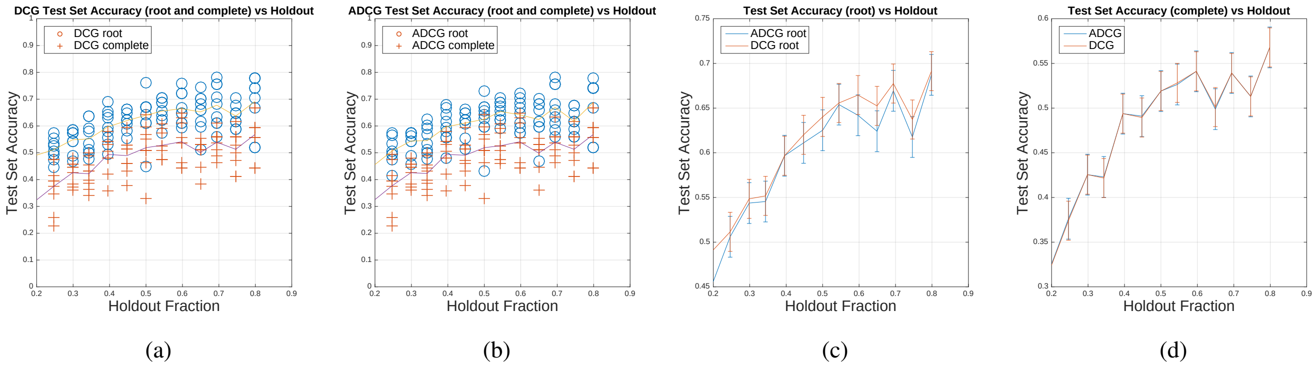
This section presents quantitative results for grounding accuracy and runtime efficiency. Further, we present qualitative examples of instructions correctly grounded as well as demonstration on the Baxter Research Robot.

A. Accuracy

We estimate the grounding accuracy for instructions in the corpus for the proposed ADCG model and the baseline DCG model, Figure 5. 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. An instruction was considered correctly grounded if (i) the root phrase was correctly grounded (root-accuracy) and (ii) all phrases in the parse tree are correctly grounded (complete-tree accuracy). The root-phrase accuracy for the ADCG model ranged between 0.361 (0.8 holdout) and 0.781 (0.3 holdout). The complete tree accuracy for the ADCG model was marginally lower ranging between 0.228 (0.75 holdout) and 0.667 (0.20 holdout). The ADCG accuracy closely followed the DCG baseline.

B. Efficiency

Table I presents the total average inference runtime normalised by the number of phrases per instruction for the corpus. The ADCG model has significantly lower average runtime than the DCG baseline. The runtime efficiency gain for the ADCG model is more pronounced with greater scene complexity. The runtime efficiency is determined by the size of the search space of groundings contributing to the number of factor evaluations. By learning a distribution over concrete groundings, the search determines a subset of highly probable concrete groundings which induce a reduced space of abstract symbols, thereby eliminating less probable factor evaluations. This approximation leads to a significant efficiency gain with minimal loss in accuracy, as demonstrated in Figure 5. The baseline model searches over the entire space of containers, as a result, scaling exponentially with with the number of concrete entities.



(a)

(b)

(c)

(d)

Fig. 5: 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 in the experiments.

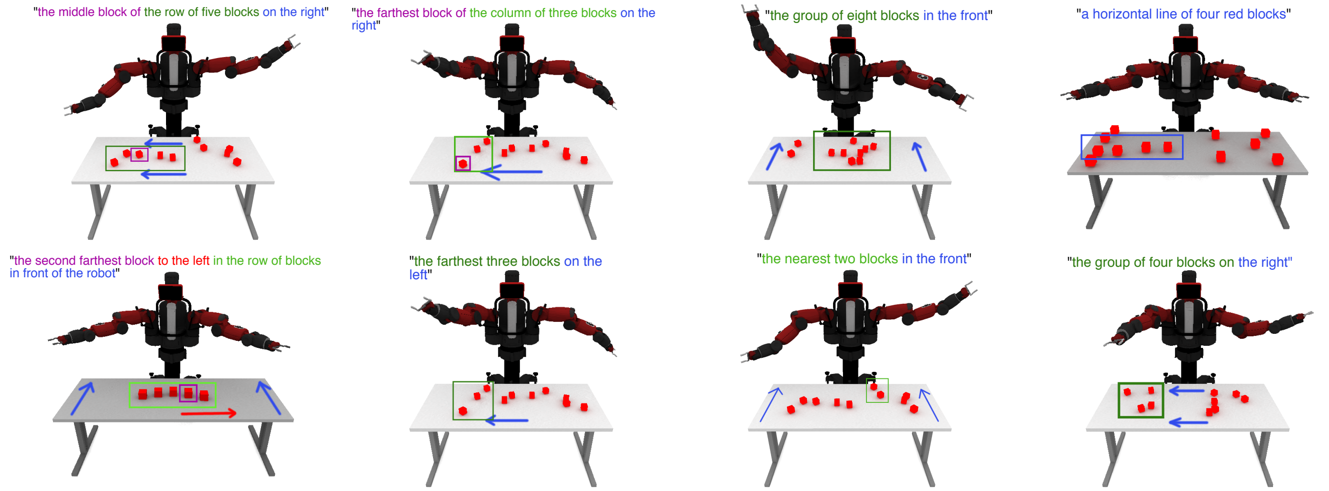

"the middle block of the row of five blocks on the right"
"the farthest block of the column of three blocks on the right"
"the group of eight blocks in the front"
"a horizontal line of four red blocks"
"the second farthest block to the left in the row of blocks in front of the robot"
"the farthest three blocks on the left"
"the nearest two blocks in the front"
"the group of four blocks on the right"

Fig. 6: Representative examples of abstract groundings that are correctly inferred by the model. The input phrases are overlaid on the images and colour coded to display constituent phrase groundings to abstract concepts, constituents and spatial relations indicated by bounding boxes and arrows. Figure best viewed in colour.

TABLE I: Average Inference Runtime for Corpus

			Runtime (seconds)	
Objects	Instructions	Worlds	DCG	ADCG
4	4	1	0.14 ±0.003	0.007 $\pm 2.3\times10^{-4}$
5	45	9	0.21 ±0.009	0.009 $\pm 5.7\times10^{-4}$
7	62	5	0.47 ±0.033	0.010 $\pm 7.9\times10^{-4}$
10	10	5	2.96 ±0.177	0.010 $\pm 1.0\times10^{-4}$
12	13	1	14.25 ±0.510	0.011 $\pm 7.2\times10^{-4}$
Total	134	21	1.89 ±4.12	0.062 $\pm 1.0\times10^{-3}$

C. Inferred Groundings

Figure 6 presents representative examples of natural language instructions correctly grounded by the model. For example, “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 constituents like “middle”, “second farthest”, “nearest” are also correctly inferred by the system. The model learns the groundings for concepts by using cues from both language cues and spatial characteristics. Figure 7 illustrates an example the input phrase, “the five blocks on the right” is inferred as an abstract container of type row, using spatial information alone in the absence of the word “row”.

D. Physical Demonstration

The system was deployed on a Baxter Research Robot. Figure 8 demonstrates the execution of the inferred actions for the natural language expressions “pick up the farthest block in the column of three blocks on the left” and“pick up the middle block of the row of five blocks on the right”. In (a), the system inferred the correct collection of blocks on the left of the effector and used the linguistic spatial cues to determine

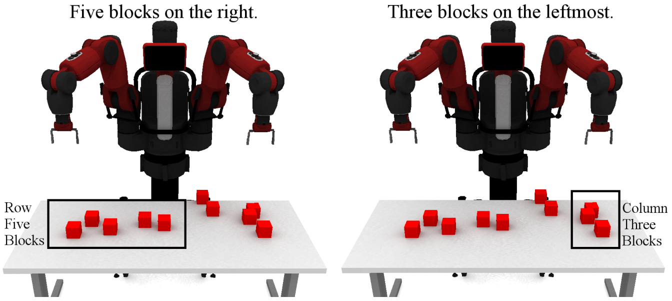
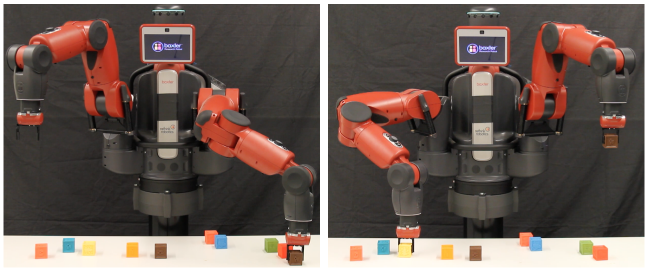


Fig. 7: Grounding examples. Left: The abstract concept "row" is inferred using spatial information in the absence of a language cue. Right: The abstract concept "column" is similarly inferred using spatial information.



(a) (b)

Fig. 8: Two images illustrating execution of correctly inferred sets of actions, objects and containers using the ADCG model for natural language expressions (a) "pick up the farthest block in the column of three blocks on the left" and (b) " pick up the middle block in the row of five blocks on the right"; in a known world setup using the Baxter Research Robot.

the correct block for grasp and vertical displacement. In (b), the system inferred the correct collection of blocks on the right hand side of the effector and determined the intended block in the centre of the collection to be grasped and displaced vertically. The language grounding step was completed in 0.33 seconds for (a) and 0.37 seconds for (b); the inferred command and grounding triggered the motion planning and task execution routine that required required 10 seconds in each case. Results were consistent across multiple runs.

VII. Related Work

The problem of grounding natural language instructions has received attention in robotics, artificial intelligence and linguistics. Ge and Mooney [10] learn a semantic parser that relates natural language descriptions to a formal representation language. Zettlemoyer and Collins [11] present an online learning approach to parse sentences into lambda-calculus representations. Chen and Mooney [12] learn a translation model for langauge using event traces from videos associated with textual commentaries. Sergio et. al. [13] demonstrate a stystem that integrates visual and spatial information with semantic parsing to interpret actions and ground spatial relations. Our approach, in contrast, is probabilistic and infers groundings related to the robot's state action space from natural language using predictors modeling spatial and parse context.

Researchers have investigated learning cost functions to model the association of language with the physical representation of the world. Kollar et. al. [14] learn cost functions that score motion verbs and spatial relations with candidate plans in the environment. Misra et. al. [15] formulate the language grounding problem as energy minimisation over a conditional random field encoding the environment and task context. Authors in [16] and [17] apply reinforcement learning to learn a mapping from language to action sequences. Matuszek et. al. [4] learn grounding relations from data that relate unconstrained language commands with control sequences in novel environments. Duvallet et. al. [18] incorporate use inverse optimal control technique to learn a cost function from language to carry out a variety of navigation behaviours. Boularias et. al. [3] use a related approach to ground language commands into a tactical behaviour specification encoding high-level behaviours. Although, these approaches can successfully learn high-level action sequences, the formulations do not express the notion of abstractions.

Our approach is based on Generalised Grounding Graphs [2] and Distributed Correspondence Graphs [1] formulations. Here, we incoporate abstract groundings and develop an efficient approximate search procedure. The work of Chung et. al. [19] is complimentary to ours. The model hierarchically infers a reduced symbol space using spatial context and input utterance but does not consider abstract groundings. Other efforts have explored grounding language by leveraging knowledge representations for reasoning about the world state [20], verifying output plans using formal logic [21], learning semantic maps using grounded natural language [22] and actively acquiring symbolic representations [23].

VIII. Conclusions

In this paper, we presented a probabilistic model for grounding natural language commands conveying abstract spatial concepts that consist of aggregations of atomic symbols in the robot's world model. Further, notions of cardinality and indexing are introduced to reference constituent elements within the aggregation. Probabilistic factors were introduced linking the expression of abstract grounding constituents with expressed concrete symbols. The model is trained using feature functions that capture spatial characteristics, language cues and child groundings. Inference is carried out via an approximate search based technique that leverages factorisation between concrete and abstract symbols and dynamically generates the search space of abstract symbols based on expressed concrete symbols, significantly pruning away less likely portions of the exponentially large grounding space. Extensive evaluation demonstrated accuracy and timing efficiency in grounding abstract concepts.

IX. 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. Authors thank anonymous reviewers and our lab colleagues for helpful advice on improving this paper.

References

[1] T. M. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6652–6659.

[2] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2011.

[3] A. Boularias, F. Duvallet, J. Oh, and A. Stentz, "Grounding spatial relations for outdoor robot navigation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1976–1982.

[4] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Procceedings of the 13th International Symposium on Experimental Robotics (ISER)*, 2012.

[5] J. Oh, A. Suppé, F. Duvallet, A. Boularias, J. Vinokurov, O. Romero, C. Lebiere, and R. Dean, "Toward mobile robots reasoning like humans," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2015, pp. 1371–1379.

[6] T. M. Howard, I. Chung, O. Propp, M. R. Walter, and N. 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.

[7] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: l-bfgs-b: fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[8] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: the penn treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[9] D. H. Younger, "Recognition and parsing of context-free languages in time n^3," *Information and Control*, vol. 10, no. 2, pp. 189–208, 1967.

[10] R. Ge and R. J. Mooney, "A statistical semantic parser that integrates syntax and semantics," in *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL)*, Ann Arbor, Michigan, 2005, pp. 9–16.

[11] L. S. Zettlemoyer and M. Collins, "Online learning of relaxed CCG grammars for parsing to logical form," in *Proceedings of Empirical Methods in Natural Language Processing and Computational Nautal Language Learning (EMNLP-CoNLL)*, Prague, 2007, pp. 678–687.

[12] D. L. Chen, J. Kim, and R. J. Mooney, "Training a multilingual sportscaster: using perceptual context to learn language," *Journal of Artificial Intelligence Research*, pp. 397–435, 2010.

[13] S. Guadarrama, L. Riano, D. Golland, D. Gouhring, Y. Jia, D. Klein, P. Abbeel, and T. Darrell, "Grounding spatial relations for human-robot interaction," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 1640–1647.

[14] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Grounding verbs of motion in natural language commands to robots," in *Experimental robotics*, Springer, 2014, pp. 31–47.

[15] D. K. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me Dave: context-sensitive grounding of natural language to manipulation instructions," in *Proceedings of Robotics: Science and Systems (RSS)*, Berkeley, USA, 2014.

[16] S. R. Branavan, H. Chen, L. S. Zettlemoyer, and R. Barzilay, "Reinforcement learning for mapping instructions to actions," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, Association for Computational Linguistics, 2009, pp. 82–90.

[17] A. Vogel and D. Jurafsky, "Learning to follow navigational directions," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguisticsi (ACL)*, Association for Computational Linguistics, 2010, pp. 806–814.

[18] F. Duvallet, M. R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz, "Inferring maps and behaviors from natural language instructions," in *Proceedings International Symposium on Experimental Robotics (ISER)*, 2014.

[19] I. Chung, O. Propp, M. R. Walter, and T. M. Howard, "On the performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2015, pp. 5247–5252.

[20] S. Lemaignan, R. Ros, E. A. Sisbot, R. Alami, and M. Beetz, "Grounding the interaction: anchoring situated discourse in everyday human-robot interaction," *International Journal of Social Robotics*, vol. 4, no. 2, pp. 181–199, 2012.

[21] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Translating structured English to robot controllers," *Advanced Robotics*, vol. 22, no. 12, pp. 1343–1359, 2008.

[22] M. R. Walter, S. Hemachandra, B. Homberg, S. Tellex, and S. Teller, "A framework for learning semantic maps from grounded natural language descriptions," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1167–1190, 2014.

[23] J. Kulick, M. Toussaint, T. Lang, and M. Lopes, "Active learning for teaching a robot grounded relational symbols.," in *IJCAI*, 2013.