# Assembling Furniture by Asking for Help from a Human Partner

Stefanie Tellex[1], Ross A. Knepper[1], Adrian Li, Thomas M. Howard, Daniela Rus, and Nicholas Roy
MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, USA

***Abstract*—Robots inevitably fail, often without the ability to recover autonomously. A human partner can help a robot recover from a failure, but it is challenging even for a willing and motivated helper to determine what actions will effectively aid the robot. To address this problem, we demonstrate an approach for enabling a robot to communicate its need for specific help to a human partner using language. Our approach automatically detects failures, then generates targeted spoken-language request such as "Hand me the white table leg." Once the human partner has repaired the failure condition, the system resumes full autonomy. We present a novel algorithm for generating help requests by emulating the human's ability to interpret a command, based on the Generalized Grounding Graph framework. We report preliminary results for an early version of our algorithm.**

## I. Introduction

We envision household robots capable of carrying out a variety of complex tasks autonomously, such as folding laundry [21], cooking dinner [1], and assembling furniture [18]. However, when robots execute these tasks autonomously, failures often occur due to perceptual errors, manipulation failures, and other issues. A key aim of current research is addressing these types of failures but eliminating them completely remains an elusive goal.

We propose an alternative approach to recovering from the inevitable failures which occur when robots execute complex tasks in real-world environments: when the robot encounters failure, it verbally asks for help from a human partner. After receiving help, it continues executing the task autonomously. As a test domain, we focus on the problem of assembling IKEA furniture, as shown in Fig. 1. We assume the robot has a pre-existing model of the piece to be assembled and a plan for assembling it. However, due to perceptual and mechanical failures, parts of the plan might be difficult or impossible to carry out. Although the human nominally directs the robot's activity, they have delegated a task to the robot and focused their attention elsewhere. Consequently, a vague request such as "Help me" does not provide enough context for the human to effectively aid the robot. Instead, it is desirable for the robot to verbally express its need with a targeted request such as "Please hand me the white table leg." The goal of our algorithm is to formulate the pithiest unambiguous natural language request so that a human not otherwise cognitively engaged can render appropriate aid.

In order to implement this strategy, the robot must first be able to detect its own failures and identify a strategy to recover from them. Next, it must communicate this strategy to the human partner. And finally, it must detect when the human has successfully or unsuccessfully provided help to the robot, in order to plan its next actions. Our algorithm generates natural language requests for help by searching for an utterance that maximizes the probability of a correspondence between the words in the language and the action the robot desires the human to perform, making use of the $G^3$ (Generalized Grounding Graph) model of a person's language understanding faculty [33]. Our hypothesis is that by modeling and bounding the probability of a human misinterpreting the request, the robot is able to generate targeted requests that work better than baselines involving either generic requests (e.g., "Help me") or template-based non-context-specific requests. We are able to compute the probability of correspondence by inverting our previous work [33], which used language to generate a graphical model that allowed us to infer the semantic meaning of the language in the robot's frame of reference. In the present work, the robot searches for language that corresponds to the action it wants the person to take. This paper presents the approach and preliminary results, but our research is ongoing, and we have not completed our experiments.

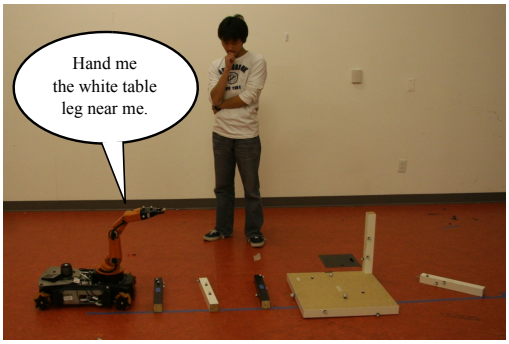



Fig. 1. When a robot needs human help with an assembly task, effective communication requires intuitive formulation of the request. Simple canned expressions like "Help me" or "Hand me white_leg_2." fail to exploit available information that could disambiguate the request. Our aim is to enable the robot to effectively communicate context-appropriate relations, such as "Hand me the white table leg near me."

## II. Related Work

In a well known-paper, Grice [13] introduces a theory of dialog expressed as maxims which participants in a conversational exchange follow in order to cooperate with each other. Clark [5] emphasizes common ground and joint activity between two participants in a dialog, which enable a conversational dyad to understand each other. Our approach aims to achieve these goals by enabling a robot to generate concise, unambiguous

[1]The first two authors contributed equally to this paper.

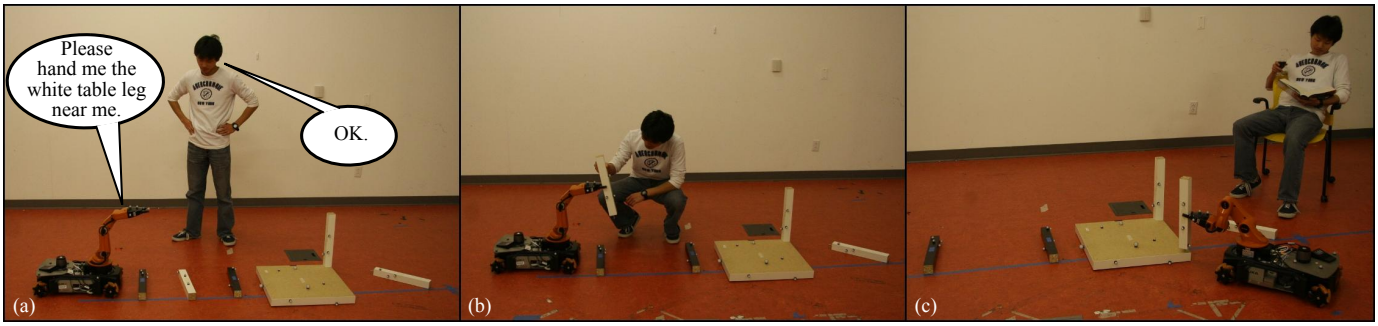



Fig. 2. During autonomous assembly, circumstances occasionally arise that the robot cannot correct. When the arrangement of parts does not permit the robot to reach its target, it may request human assistance (a). After this brief human intervention (b), autonomous operation resumes (c).

natural language requests by semantically modeling the listener's ability to understand them.

Traditional methods for generating language rely on a dedicated language-generation system that is not integrated with a language-understanding framework [17, 25]. Striegnitz et al. [31] give an overview of the GIVE challenge, a language generation task in which systems automatically generate instructions for humans moving through virtual environments; one typical system is described by Garoufi and Koller [11]. These approaches typically consist of a sentence planner combined with a surface realizer to guide decision making of what to say, but contain no principled model of how an instruction-follower would comprehend the instruction. Our approach differs in that it inverts a module for language understanding in a principled way.

Some previous work has approached the generation problem by inverting a semantics model. Golland et al. [12] use a game-theoretic approach combined with a semantics model to generate referring expressions. Roy [28] presents an algorithm for generating referring expressions in a two-dimensional geometric scene which uses an ambiguity score to assess the quality of candidate descriptions in a two-dimensional domain. Chen et al. [4] describe a system that learns to generate language describing robot soccer games. Our algorithm, in contrast, generates complete commands rather than noun phrases and asks the listener to follow a complex command rather than simply selecting an object.

Our approach views the language generation problem as inverse language understanding, building on the $G^3$ approach described by Tellex et al. [33]. A large body of work focuses on language understanding for robots [9, 19, 20, 22]. The $G^3$ framework particularly lends itself to inversion because it is a probabilistic framework which explicitly models the mapping between words in language and aspects of the external world, so metrics based on entropy may be used to assess the quality of generated utterances.

Robotic assembly of complex objects has been explored in isolation, including planning [2, 6, 15, 26] and control [3, 16, 24, 27]. Complete assembly systems have largely been targeted at the space domain due to the adverse conditions for supporting human life in orbit and on other planets [7, 8, 24, 29, 30, 32]. Such approaches can assume that human partners, if present, are highly trained astronauts. Less work has targeted assembly tasks for untrained users, such as in the home environment.

Cooperative human-robot activities, including assembly, have been broadly studied. Ogata and Takahashi [23] explore a visual virtual interface enabling a human to teach a robot to perform an action. Wilson [34] explores automated planning with human help. Systems presented by Simmons et al. [29] and Dorais et al. [8] perform assembly in the space domain with human intervention for failure recovery. The architectures permit various granularities of human intervention through a sliding autonomy framework. A failure triggers the replay of video of the events preceding failure, from which the human must glean situational awareness. In contrast, our approach leverages natural language to convey to the user exactly how the problem should be resolved.

Fong et al. [10] explore an interface for robot-assisted teleoperation with robot-initiated dialog to help avoid or resolve various failures. Dialog is implemented through a text-based modal interface on a handheld device. Similar to our approach, the authors treat the human as a resource for the robot, interfaced through natural language. However, Fong et al. use fixed templates to convey information, whereas our focus is on maximizing comprehension and situational awareness through flexible language choice.

## III. Assembling Furniture With Robots

Knepper et al. [18] describe the complete system for furniture assembly used in this paper. This framework consists of a team of youBots which collaborated to autonomously assemble a piece of furniture. The robotic team often succeeds but occasionally encounters failures which require human intervention. In this paper we explore strategies for enabling a human to render assistance to the robot team.

This section gives an overview of the assembly system and describes how we have augmented it with the ability to recognize a failure condition and generate a symbolic (non-linguistic) solution to the problem that has occurred. This symbolic request will form the input to the language generation system. Since perception is not a focus of this paper, we employ a VICON motion capture system to track the location of each participating robot, human and furniture part throughout the course of the assembly process.

A team of robots receives assembly instructions encoded in ABPL (A Better Planning Language) [18]. The ABPL

symbolic planner is implemented as a wrapper around Fast-Forward [14], an off-the-shelf symbolic planning package. In this paper, we focus on the example of the IKEA Lack table, which comprises one table top and four legs. The hand-coded ABPL specification for its assembly involves twelve actions. The symbolic planner requires two seconds to return a solution of 48 steps for two robots using both their native grippers and a custom screwing tool to assemble and reorient the table.

Although the robots are capable of assembling the table in parallel, we employ a centralized assembly executive in this work to avoid the situation in which several robots require human assistance at the same time. The centralized executive takes as input the symbolic plan and executes each plan step in sequence. Each symbolic action corresponds to a manipulation or perception action to be performed by one or two robots. Examples include `locate_part`, `navigate_to_part`, `pick_up_leg`, `hand_off_table_leg`, and `attach_table_leg`. Execution of the 48-step plan takes approximately ten minutes when no failures occur.

### A. Detecting Failures

To detect and address failures, the system tracks and compares the states of the symbolic plan and the external world. A robot can recognize that a failure has occurred when the internal state of the assembly executive, $q_s$, does not match the state of the external world, $q_w$. The fluent state vectors $q_w$ and $q_s$ are binary vectors where each entry corresponds to a predicate calculus expression. Following the execution of each action on the robot, the system updates the internal state $q_s$ according to the post-conditions of that action. Thus, $q_s$ can only change with the knowledge of the assembly executive.

The external symbolic state, $q_w$, is computed from the tracked pose of every rigid body known to the VICON system, including each furniture part, each robot chassis and hand, and each human. The VICON state, $x_w \in \mathbb{R}^n$, is continuous and high-dimensional. A function $f$ maps $x_w$ onto the lower-dimensional vector $q_w$, with the same interpretation as $q_s$. The function $f$ is fast to evaluate and is suitable for frequent execution during the course of assembly. At present, it is coded by hand based on the furniture assembly context. Automatically learning $f$ remains a subject for future work. The external state, $q_w$, may change independently of any deliberate robot action, such as by human intervention or as an unintended side-effect of some robot action.

The two fluent vectors, $q_s$ and $q_w$, can each be used to verify the truth of pre- or post-conditions at any time. During the course of executing a symbolic plan, the symbolic planner guarantees that any such condition is trivially consistent with $q_s$. Prior to executing each action, the assembly executive verifies the action's preconditions against $q_w$. Likewise, following each action, the post-conditions are similarly verified. Any unsatisfied condition indicates a failure and triggers the assembly executive to pause the assembly process and initiate error recovery.

### B. Recovery Strategy

When a failure occurs, its description takes the form of an unsatisfied condition. Example failure conditions include

- ¬visible(table_leg_2),
- ¬arm_holding(robot0, table_leg_2), and
- ¬attached_to(table_leg_2, white_table_top_hole1).

Such unsatisfied expressions do not inherently contain a remedy. We regard the source of initiative for communicating the problem and generating an appropriate remedy as an experimental variable in this work. Various approaches considered in this paper span a spectrum of initiative, aiming for a robot that can work autonomously some of the time, but rely on human help when it encounters failure.

.

One solution, which places all initiative on the robot, is to assign $q_s = q_w$ and rerun the symbolic planner from this new initial state. The likely outcome is that the symbolic planner will retry the failed action. We refer to this method as Approach 0. This solution is not complete since some failures could result in infinite retry loops.

A second approach is for the robot to ask the human to address the problem. The robot first computes actions that, if taken, would resolve the failure and enable it to continue assembling the piece autonomously. The system computes these actions by processing failed conditions. At present, a single symbolic action is hard-coded for each commonly-observed failure. Remedy requests are expressed in a simple symbolic language. This symbolic request, $a$, specifies the action that the robot would like the person to take to help it recover from failures. The symbolic language consists of the following types of requests:

- `align_with_hole(leg, table)`
- `give_part(robot, leg)`
- `pick_up(leg)`
- `put_near(leg)`
- `locate_part(leg)`
- `screw_in_table_leg(leg, hole)`

However these symbolic forms are not appropriate for speaking to an untrained user. In the following section, we explore a series of approaches that take as input the symbolic request for help and generate a language expression asking a human for assistance.

## IV. Asking for Help From a Human Partner

Our aim is to find the words $\Lambda$ which effectively communicate the robot's desired action $a$ to an untrained human. Our framework generates a request for help by inverting a model of natural language semantics. We use the $G^3$ semantics model to identify sentences that unambiguously specify the action the robot wishes the human to take. The $G^3$ model imposes a distribution over *groundings* in the external world, $\gamma_1 \ldots \gamma_N$, given a natural language sentence $\Lambda$. Groundings are the specific physical concepts that are referred to by the language and can be objects (e.g., a truck or a door), places (e.g., a particular location in the world), paths (e.g., a trajectory through the environment), or events (e.g., a sequence of actions taken by the robot). Each grounding corresponds to a particular constituent $\lambda_i \in \Lambda$. For example, for a sentence such as "Pick up the table leg," the grounding for the phrase "the table leg" corresponds to an actual table leg in the external world, and the grounding for the entire sentence corresponds

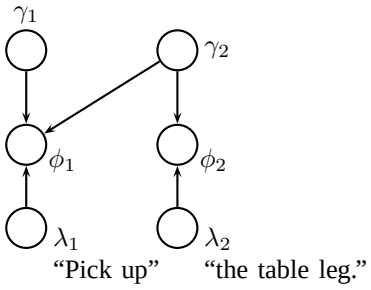



Fig. 3. Grounding graph for the command, "Pick up the table leg." Random variables and edges are created in the graphical model for each constituent in the parse tree. The $\lambda$ variables correspond to language; the $\gamma$ variables correspond to groundings in the external world. Edges in the graph are created according to the parse structure of the command

to the trajectory of a person as they execute the command. Understanding a sentence in the G$^3$ framework amounts to the following inference problem:

$$\underset{\gamma_1 \ldots \gamma_N}{\operatorname{argmax}} \, p(\gamma_1 \ldots \gamma_N | \Lambda, M) \tag{1}$$

The environment model $M$ consists of the robot's location along with the locations and geometries of objects in the external world. A robot computes the environment model using sensor input; in the domain of furniture assembly, the system creates the environment model using input from VICON. The computed environment model defines a space of possible values for the grounding variables, $\gamma_1 \ldots \gamma_N$.

To invert the model, the system converts the symbolic action request $a$ to the grounding variable, $\gamma_a$, which corresponds to the entire sentence. It then searches for the most likely sentence $\Lambda$ given that variable according to the semantics model:

$$\underset{\Lambda}{\operatorname{argmax}} \, p(\Lambda | \gamma_a, M) \tag{2}$$

Maximizing this objective is equivalent to maximizing the joint distribution:

$$\underset{\Lambda}{\operatorname{argmax}} \, p(\Lambda, \gamma_a, M) \tag{3}$$

Since $\Lambda$ may consist of more than one linguistic constituent, we need to introduce $\gamma_i$ variables for each constituent, which we marginalize out:

$$\underset{\Lambda}{\operatorname{argmax}} \sum_{\gamma_1 \ldots \gamma_{a-1}, \gamma_{a+1} \ldots \gamma_N} p(\Lambda, \gamma_1 \ldots \gamma_N, M) \tag{4}$$

To factor the model, we introduce a correspondence vector, $\Phi$, as in Tellex et al. [33]. Each entry $\phi_i \in \Phi$ corresponds to whether linguistic constituent $\lambda_i \in \Lambda$ corresponds to the groundings associated with that constituent. For example, the correspondence variable would be $True$ for the phrase "the white table leg" and a grounding of a white leg, and $False$ if the grounding was a different object, such as a black table top. We assume that $\gamma_1 \ldots \gamma_N$ are independent of $\Lambda$ unless $\Phi$ is known. Introducing $\Phi$ enables factorization according to the structure of language with local normalization at each factor over a space of just the two possible values for $\phi_i$.

The optimization then becomes:

$$\underset{\Lambda}{\operatorname{argmax}} \sum_{\gamma_1 \ldots \gamma_{a-1}, \gamma_{a+1} \ldots \gamma_N} p(\Phi | \Lambda, \gamma_1 \ldots \gamma_N, M) \tag{5}$$

We factor the expression according to the compositional syntactic structure of the language $\Lambda$. This factorization can be represented as a directed graphical model where random variables and edges in the model are created according to the structure of the language. We refer to one of these graphical models as a *grounding graph*. The details of the factorization are described by Tellex et al. [33].

$$\underset{\Lambda}{\operatorname{argmax}} \sum_{\gamma_1 \ldots \gamma_{a-1}, \gamma_{a+1} \ldots \gamma_N} \prod_i p(\phi_i | \lambda_i, \gamma_1 \ldots \gamma_N, M) \tag{6}$$

We can compute the inner term in Equation 6 directly using the G$^3$ framework, as described in Tellex et al. [33]. Briefly, each factor is a log-linear model whose features are trained from a labeled corpus of language aligned with corresponding groundings in the external world.

To train the model, we collected a new dataset of natural language commands given by a human to another human in the furniture assembly domain. We created twenty-one videos of a person executing a task involved in assembling a piece of furniture. For example, one video showed a person screwing a table leg into a table, and another showed a person handing a table leg to a second person. The people and objects in the video were tracked with VICON so each video has an associated context consisting of the locations, geometries, and trajectories of the people and objects. We asked annotators on Amazon Mechanical Turk to view the videos and write a natural language command they would give to ask one of the people to carry out the action depicted in the video. Then we annotated commands in the video with associated groundings in the VICON data. The corpus contains 326 commands with a total of 3279 words.

We carry out the inference in Equation 6 by generating candidate sentences using a context-free grammar (CFG). Our CFG defines a structured search space for the $\Lambda$ variable and appears in Figure 4. The inference procedure creates a grounding graph for each candidate sentence using the parse structure derived from the CFG and then scores it according to Equation 6. Marginalizing over the grounding variables takes significant computation, so we approximate the summation using high-probability values found using beam search.

Since even with a carefully coded grammar the search space is large, we are exploring heuristics for quickly scoring candidate sentences. We plan to use a discriminative probabilistic model that approximates the likelihood of a particular parse structure given the symbolic request and features of the environment. The model is trained using features that capture the influence of the environment such as the presence of objects that are similar in appearance. Our hypothesis is that when there is more ambiguity in the environment, there will be a bias towards more complex syntactic structures which can uniquely specify the desired action.

For comparison, we report the performance of two baselines. The simplest approach from the assembly executive's perspective is to delegate diagnosis and solution of the problem

$$
\begin{aligned}
S &\rightarrow VB \;\; NP \\
S &\rightarrow VB \;\; NP \;\; PP \\
PP &\rightarrow TO \;\; NP \\
VB &\rightarrow \text{align|give|hand|lift|pick up|place|put|screw} \\
NP &\rightarrow \text{the white leg|the black leg|me|the leg|} \\
&\quad\;\; \text{the hole|the white table|the black table} \\
TO &\rightarrow \text{above|by|near|under|with}
\end{aligned}
$$

Fig. 4. Part of the context-free grammar defining the linguistic search space.

| | |
|---|---|
| Symbolic request | `give_part(robot3, white_leg_0)` |
| "Help me" baseline | "Help me." |
| Template baseline | "Please hand me white leg 0." |
| $G^3$ | "Hand me the white leg." |

Fig. 5. Initial scene from our dataset, the command in the symbolic language, and the instructions generated by each approach.

to the human with the simple fixed request, $\Lambda =$ "Help me." Whereas this inference might be easy for the system designer, it is often very challenging for an untrained user who does not understand the inner workings of the robotic system.

A second baseline delegates initiative for diagnosis of the problem to the programmer in advance by constructing a look-up table of templates comprising commands to a human helper, similar to the approach of Fong et al. [10] among others. These generic requests take the following form:

- "Place table leg 2 where I can see it,"
- "Put table leg 2 in robot 0's hand," and
- "Attach table leg 2 at location 1 on the table top."

## V. Experimental Results

Our experimental work is ongoing. We provide a preliminary report on our evaluation plan and initial results. We plan to assess the performance of the system by performing a quantitative corpus-based evaluation. Our evaluation consists of a dataset of conditions in which the robot generates a request for help. In each scene, the robot has a partially assembled piece of furniture and asks for human assistance to complete the next step in its task. We manually generated five different help requests in the symbolic language based on typical failure modes for the robot. Then we filmed an actor responding to the request. This paradigm results in several different actions being filmed for each initial condition, corresponding to each symbolic request for help. The four scenes appear in Figure 6, together with the natural language help requests intended by the authors. The positions and geometries of the objects in the scene were tracked with VICON, as well as the movements of the actor. Finally, we annotated each video with a symbolic request for help as described Section III-B.

TABLE I
Fraction of Correctly Followed Commands

| Metric | % Success | |
|---|---|---|
| Chance | 20 | |
| "Help me" Baseline | 21 | $\pm 8.0$ |
| Template Baseline | 64 | $\pm 9.4$ |
| $G^3$ Inverse Semantics (no graph search) | 50 | $\pm 9.8$ |
| Oracle | 100 | |

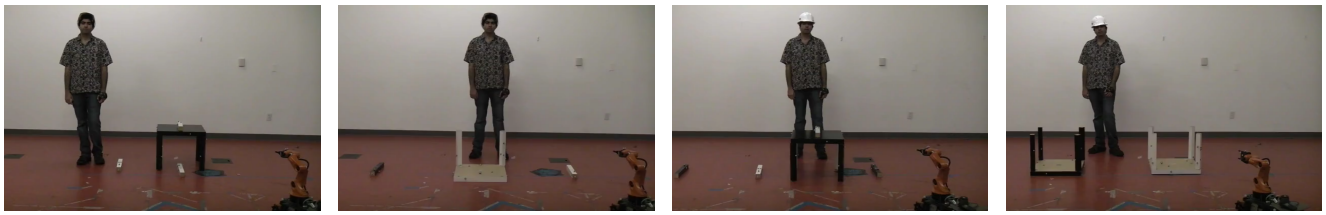
We automatically generated requests for help for each scenario in the evaluation using several different methods. The input is the locations and geometries of objects in the scene, as well as the symbolic request for help. The output is a natural language request for help, using one of the approaches from Section IV. Our implementation of the $G^3$ inverse semantics model is incomplete and searches through a hardcoded set of graphical structures. The results exhibited for the $G^3$ model assumed a single sentence structure determined by the type of symbolic request. Figure 5 shows an initial scene together with a request for help in the symbolic language and language from each of the three approaches.

To assess the performance of each method, we showed annotators on Amazon Mechanical Turk the generated command, along with the five videos generated for each initial scene. We asked annotators to identify the video that is most closely shows the action they would perform in response to the natural language command. This evaluation enables us to identify which commands generate understandable instructions that uniquely specify which actions the human should take to best assist the robot. Results appear in Table I. We report 95% confidence intervals. Chance performance is 20%. Each of the twenty generated commands was issued to five different subjects, for a total of 100 trials. Seventeen subjects participated in the study.

Our initial results show that the "Help me" baseline performs at chance, whereas the Template baseline and the $G^3$ inverse semantics model both improve performance significantly. Perfect performance is 100%, accurately choosing the correct video for each of the twenty tasks. The inverse semantics model using $G^3$ is outperformed by the Template baseline. This result is not surprising since this initial implementation does not search over the space of different syntactic structures. We hypothesize that as we add the CFG and discriminative model the richer commands will outperform the template baseline. We are actively continuing our experiments and plan to add additional results for the inverse semantics model.

## VI. Discussion and Future Work

This work represents a step toward the goal of mixed-initiative human-robot cooperative assembly. Our aim is to create an algorithm for enabling a robot to generate natural language help requests. A human teammate can provide targeted help to the robot without requiring detailed situational



(a)
Hand me the leg on the table.
Hand me the leg under the table.
Hand me the leg near the table.
Pick up the leg on the table.
Put the leg on the table near me.

(b)
Screw in the white leg.
Screw in the black leg.
Place the white leg in the hole.
Place the white leg near the table.
Put the white table near the white leg.

(c)
Bring me the white leg near the table.
Bring me the black leg near the table.
Bring me the leg far from the table.
Hand me the white leg on the table.
Pick up the white leg near the table.

(d)
Flip the white table.
Flip the black table.
Put the white table near me.
Pick up the white table.
Hand me the white table.

Fig. 6. The four initial scenes from the evaluation dataset, together with the manually-generated help requests

awareness of the robot and its failure modes. Our technical approach is to invert a model of semantics and use it to generate natural language requests that are easy for a human to understand.

After completing our initial experiments, we plan to extend our approach to inverse semantics by evaluating a command based on the entropy over actions it produces according to the semantics model: good commands are ones with low entropy over the resulting distribution of actions that that human partner should take. By modeling the listener's uncertainty, our hypothesis is that the robot can generate high-quality commands that enable it to recover from failure. A forthcoming complexity analysis will demonstrate the feasibility of the approach.

Our framework could be extended to support a variety of failure recovery actions, so that the robot can choose between natural language actions and other types of non-social actions that may gather additional information or resolve the problem in other ways. Such an approach would assign a cost to each action that incorporates factors such as a human's capabilities in relation to the robots needs, the social cost of disturbing a person, and the time and energy required by the action. Each cost should be weighted by an estimate of the probability of successfully resolving the failure by each action. Over time, the robot would then learn which situations are most appropriate for seeking human intervention. In order for such a model to evaluate the efficacy of a human's attempt at assistance, the robot must be capable of detecting human failures, which requires understanding of human intent.

As we move from robot-initiative to mixed-initiative communication, the reliance on common ground and context increase significantly. Since our models can be expected to remain imperfect, the demand for unambiguous sentences becomes less satisfiable. We could tolerate increased ambiguity on the part of both the human and robot by engaging in back-and-forth dialog between the two. During conversation, recent utterances serve to identify salient aspects of the environment and situation, thus simplifying the inference problem.

In the long term, we aim to develop robots with increased task-robustness in a variety of domains by leveraging the ability and willingness of human partners to assist robots in recovering from a wide variety of failures.

## VII. Acknowledgments

This work was supported in part by the Boeing Company, and in part by the U.S Army Research Laboratory under the Robotics Collaborative Technology Alliance.

## References

[1] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus. Interpreting and executing recipes with a cooking robot. In *13th International Symposium on Experimental Robotics*, 2012.
[2] A. Bourjault. *Contribution a une approche méthodologique de lassemblage automatisé: Elaboration automatique des séquences opératiores*. PhD thesis, LUniversité de Franche-Comté, Nov 1984.
[3] F. Caccavale, C. Natale, B. Siciliano, and L. Villani. Achieving a cooperative behavior in a dual-arm robot system via a modular control structure. *Journal of Robotic Systems*, 18(12):691–699, 2001.
[4] David L Chen, Joohyun Kim, and Raymond J Mooney. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37(1):397–436, 2010.
[5] Herbert H. Clark. *Using Language*. Cambridge University Press, May 1996. ISBN 0521567459.
[6] T. L. De Fazio and D. E. Whitney. Simplified generation of all mechanical assembly sequences. *IEEE Journal of Robotics and Automation*, RA-3(6), Dec. 1987.
[7] W. Doggett. Robotic assembly of truss structures for space systems and future research plans. In *Proceedings of the IEEE Aerospace Conference*, volume 7, 2002.
[8] G. Dorais, R. Banasso, D. Kortenkamp, P. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on mars, 1998.
[9] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn. 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*, pages 4163–4168, 2009.
[10] T. Fong, C. Thorpe, and C. Baur. Robot, asker of questions. *Journal of Robotics and Autonomous Systems*, 42: 235–243, 2003.
[11] K. Garoufi and A. Koller. 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, 2011.
[12] D. Golland, P. Liang, and D. Klein. A game-theoretic approach to generating spatial descriptions. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 410–419. Association for Computational Linguistics, 2010.
[13] H. P. Grice. *Logic and Conversation'in P. Cole and J. Morgan (eds.) Syntax and Semantics Volume 3: Speech Acts*. Academic Press, New York, 1975.
[14] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14:253–302, month 2001.
[15] L.S. Homem de Mello and A.C. Sanderson. A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Transactions on Robotics and Automation*, 7(2):228–240, April 1991.
[16] H. Inoue. Force feedback in precise assembly tasks. Technical Report 308, Massachusetts Institute of Technology, Artificial Intelligence Lab, 1974.
[17] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Pearson Prentice Hall, 2 edition, May 2008. ISBN 0131873210.
[18] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus. IkeaBot: An autonomous multi-robot coordinated furniture assembly system. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013.
[19] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proc. ACM/IEEE Int'l Conf. on Human-Robot Interaction*, pages 259–266, 2010.
[20] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proc. Nat'l Conf. on Artificial Intelligence (AAAI)*, pages 1475–1482, 2006.
[21] J. Maitin-Shepard, J. Lei, M. Cusumano-Towner, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Anchorage, Alaska, USA, May 2010.
[22] C. Matuszek, N. FitzGerald, L. Zettlemoyer, L. Bo, and D. Fox. A joint model of language and perception for grounded attribute learning. *Arxiv preprint arXiv:1206.6423*, 2012.
[23] H. Ogata and T. Takahashi. Robotic assembly operation teaching in a virtual environment. *IEEE Transactions on Robotics and Automation*, 10(3):391–399, Jun 1994.
[24] F. Ozaki, K. Machida, J. Oaki, and T. Iwata. Robot control strategy for in-orbit assembly of a micro satellite. *Advanced Robotics*, 18(2):199–222, 2004.
[25] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Cambridge University Press, January 2000. ISBN 9780521620369.
[26] J. Latombe R.H. Wilson. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71:371–396, 1994.
[27] J. Rojas and R. A. Peters, II. Analysis of autonomous cooperative assembly using coordination schemes by heterogeneous robots using a control basis approach. *Autonomous Robots*, 32(4):369–383, 2012.
[28] D. Roy. A trainable visually-grounded spoken language generation system. In *Proceedings of the International Conference of Spoken Language Processing*, 2002.
[29] R. Simmons, S. Singh, F. Heger, L. M. Hiatt, S. C. Koterba, N. Melchior, and B. P. Sellner. Human-robot teams for large-scale assembly. In *Proceedings of the NASA Science Technology Conference*, May 2007.
[30] P. J. Staritz, S. Skaff, C. Urmson, and W. Whittaker. Skyworker: A robot for assembly, inspection, and maintenance of large scale orbital facilities. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Seoul, Korea, May 2001.
[31] K. Striegnitz, A. Denis, A. Gargett, K. Garoufi, A. Koller, and M. Theune. Report on the second second challenge on generating instructions in virtual environments (give-2.5). In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 270–279. Association for Computational Linguistics, 2011.
[32] A.W. Stroupe, T. Huntsberger, B. Kennedy, H. Aghazarian, E.T. Baumgartner, A. Ganino, M. Garrett, A. Okon, M. Robinson, and J.A. Townsend. Heterogeneous robotic systems for assembly and servicing. In *Proceedings of the International Symposium on Artifical Intelligence, Robotics and Automation in Space*, Munich, Germany, August 2005.
[33] S. Tellex, T. Kollar, S. Dickerson, M.R. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proc. AAAI*, 2011.
[34] R.H. Wilson. Minimizing user queries in interactive assembly planning. *IEEE Transactions on Robotics and Automation*, 11(2), April 1995.