Interactive Hierarchical Task Learning via Crowdsourcing for Robot Adaptability

Aaron St. Clair*, Carl Saldanha*, Adrian Boteanu†, and Sonia Chernova*

*School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA.
† Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, USA.
Email: astclair@cc.gatech.edu, csaldanha3@gatech.edu, ab2633@cornell.edu, chernova@cc.gatech.edu

Abstract—This paper describes the application of crowdsourcing to the problem of interactive task learning with the aim of enabling remotely-located users to effectively teach robots to perform complex tasks in real-world environments. We present a novel system that allows users to demonstrate hierarchical tasks via web-based control of a table-mounted six degree of freedom robot arm. The system employs intelligent action grouping suggestions and substitution suggestions for error recovery to assist the user in providing quality demonstrations. In addition, we describe design considerations and proposed extensions for the effective application of crowdsourcing to several human-robot interaction scenarios as motivated by this initial study.

I. INTRODUCTION

For robots to be effective assistants in a wide variety of everyday environments working alongside humans they must be capable of adapting to new settings and learning new skills and tasks as needed with minimal overhead. The primary goal of this work is to enable remote, non-expert users to teach a robot a complex task, in this case involving several sequenced object manipulation subtasks. Through this process, we envision a robot that can be adapted to novel environments, taught new skills after deployment, and trained to provide better cues and feedback with people in its environment.

Crowdsourcing and web robotics provides a promising platform for collecting these types of demonstrations, where users do not need to physically interact with the robot or the environment and no hard timing requirements exist on the execution of actions. Web-based crowdsourcing is potentially more amenable to large scale data collection than in-person face-to-face interactions where issues such as participant safety and recruitment can be barriers to accruing large numbers of participants. Crowdsourcing also allows for the teaching process to be spread over multiple participants, each providing unique demonstrations, and also allows for multiple users to interact with the system at a given time, potentially providing additional metadata or evaluation of other users.

The unique constraints imposed by a web-based data collection also presents a number of challenges for the design of learning systems and associated interfaces. These include providing users with enough situational awareness to effectively operate the robot, issues with anonymous users such as removing malicious users and ensuring quality demonstrations, as well as methods for combining a potentially large number of demonstrations to produce coherent robot task behavior.

II. CROWDSOURCED HIERARCHICAL TASK LEARNING

Our system consists of a web-based user interface implemented using the Robot Management System (RMS) framework as detailed by Toris et al. [18]. Using the interface remote users with an internet connection and a compatible browser control a 6 degree of freedom Kinova Jaco robot arm fixed to a table. Since we are primarily concerned with pick-and-place tasks we assume the robot can directly execute two actions: pickup, i.e., pickup an object on the table, and store i.e., drop an item in the gripper into one of several containers. These primitive actions are in turn built upon another crowdsourced system for generating object models and reliable grasp points from user demonstrated grasp poses as described in Kent et al. [9] and Kent and Chernova [8].

Learning from demonstration (LfD) [1] is a well-studied technique allowing a person to teach a robot a new task by demonstrating the behavior. There is prior work involving both robots and virtual agents on hierarchical task learning: [5, 14, 19, 6, 16], interactive task learning: [3, 4] and learning from a single task iteration: [7, 12]. In our system, users teach the robot a new task represented by a hierarchical task network (HTN), a tree representation of the task structure with both primitive actions that the robot can execute directly and abstract actions that consist of a sequence of other actions to perform. These primitive actions are sequenced and grouped as the user executes a demonstration of the desired new behavior using a question-and-answer interaction as described.
in Mohseni-Kabir et al. [13]. For example, if a user executes
the Pickup action and then the Store action the system will
ask to group the actions into a new abstract action based on
common input and output parameters. The hierarchical nature
of HTNs provides a mechanism for users to reuse abstract
tasks to accomplish more complex series of primitive actions
and provides a natural means for providing communicative
feedback to the user on the status of the task.

A. Error Recovery via Object Substitution

During the course of the demonstration, tasks may fail to
occur as the robot performs the actions requested by the user.
If an item required to perform a given action is not present
at execution time, perhaps due to being used in an earlier
portion of the task or being missing altogether, we employ an
error recovery mechanism that intelligently suggests substitute
objects as described in Boteanu and Chernova [2].

We suggest object substitutions for missing objects using
bag-of-words contexts derived from the HTN actions in con-
junction with existing semantic networks. A semantic network
represents concepts as vertices of a graph and relations be-
tween these concepts as edges. We used two such resources
to derive concept similarity: WordNet and ConceptNet. WordNet
represents word senses by associating concepts with
synsets – different senses of a word belong to different synsets.
It provides a concept similarity measure as the normalized
path distance [15]. On the other hand, ConceptNet aggregates
data from a variety of sources, including WordNet, DBPedia,
and crowd-contributed information [11]. It covers a broader
spectrum words, and represents a total of 48 relations such as
part-of, is-a, used-for, has-property.

Because it aggregates information, ConceptNet contains
some noise in both the nodes (e.g. different spellings) and
the edges (e.g. incorrect edges). Using the ConceptNet graph,
Divisi is a measure of word similarity which uses its singular
value decomposition [17]. In addition to these similarity mea-
ures, we use the Semantic Similarity Engine (SSE) to evaluate
relational similarity [2]. SSE computes the most similar path
pair between two pairs of words using ConceptNet, producing
a numerical normalized value, as well as a human readable
justification. SSE targets proportional analogies, which are
commonly phrased as A is to B as C is to D. For object
substitution, we can leverage this type of analogy to attempt
to establish relational parallelism between a target and a
candidate with respect to a context element, by forming the
analogy target:context word::candidate:context word.

The system we have implemented uses active object sub-
stitution to recover from errors due to missing objects. In
practice this enables greater re-use and adaptation of abstract
tasks. In our initial study users are asked to pack a set of
lunches with food items present on the table. The instructions
are to pack each lunch with a balanced mix of items from
each category and users begin with only two primitive actions
Pickup(object) and Store(object, container). Remote
users have successfully used the interface to teach the lunch-
packing task to the robot based. We are currently in the process
of conducting a large-scale user study to determine the affects
of the object substitution error recovery method on the number,
quality, and speed of crowdsourced task demonstrations.

III. EXTENSIONS TO CROWDSOURCED TASK LEARNING

In this section we describe several extensions to the problem
of crowdsourcing interactive task learning including user en-
agement management and leveraging multiple users to collect
more and/or richer data. The current experiment allows users
to control a real robot arm but the interface could also be
used with a simulated robot, allowing multiple users to interact
with the interface. This would increase the scalability of the
data collection ability but depending on the fidelity of the
simulator may sacrifice some reproducibility with the target
robot system. For higher-level task planning problems a lower
fidelity simulation may suffice, while some tasks such as grasp
demonstrations likely require higher fidelity.

One concern with online users is ensuring they are mo-
tivated to perform the task and remain engaged even if the
robot encounters errors or is slower than expected to complete
task. Unlike a user brought into the lab to teach the task
in person, where the novelty effect of interacting with the
robot may be reward enough, online users may be more easily
distracted or apt to quit. Gamification of the task by adding
elements to the interface such as score-keeping may affect
user engagement. Additionally, in rehabilitation scenarios,
another potentially tedious task setting, some results have
demonstrated that knowledge of results, or telling people how
did, and social-comparative feedback, or telling people
how their performance compared to others, may have positive
effects. Additional investigation is needed to determine what
factors affect users engagement during teaching.

If a large user community is built-up around the crowd-
sourcing effort, there is the potential to leverage multiple users
visiting the site at the same time. First, if the hardware required
for teaching the task is available in sufficient numbers the sys-
tem can parallelize task demonstrations by queuing users and
assigning them to the first available robot. Beyond this there
is also the potential to utilize multiple users interacting with a
single system to provide richer teaching data. One technique
is to partition the problem into smaller subproblems that can
be assigned to different users as in prior work by Lasecki et al.
[10]. The object modeling and grasp demonstration interface
used for the primitive actions [9] uses this method, where
users each provide a subset of the grasp data used to build
the resulting object models and unified grasp points for each
object. These methods work well when a problem can be
naturally decomposed into a series of smaller subproblems
that can be solved individually. In large, complex tasks, for
instance, this technique could allow for demonstrations of
abstract actions that accomplish subtask goals.

Another method is to augment the single-user interface by
allowing other users to view the progress of the user in direct
control. This setup is promising in that the queued users can
provide metadata about the active user’s performance. They
might for instance be asked to suggest alternate recipes for
achieving a desired outcome, to annotate completion conditions for a given subtask, to provide natural language descriptions of actions, or to identify the temporal dependencies of task actions. These types of metadata could be leveraged when the new task behavior is deployed and the robot must execute the task behavior around or with the help of a person. The users monitoring the active user may also be able to provide help and guidance through a chat interface and could also potentially identify malicious users if the user community is attentive and invested in the outcome of the experiment. Finally, by monitoring the behavior of the queued users we may be able to elicit better teaching behavior by reordering the queue to prefer more actively engaged users. Additional investigation is necessary to determine what aspects of queue behavior are predictive of higher-quality teachers. We have implemented a user queue management mechanism and will be applying it to these scenarios in future work.

IV. CONCLUSION

We described a system which crowdsources robot task specification and execution to non-expert, remote users via a web-based interface. The user builds a complex, hierarchical task by executing and grouping primitive actions. The interface prompts users to group items as needed, reducing the amount of actions the user must execute. The system also employs error recovery behaviors that support adapting abstract actions by dynamically suggest object substitutions to mitigate discrepancies between the task definition and the environment. Additionally, we describe a series of extensions in applying crowdsourcing in interactive task learning domains.

REFERENCES


