Incremental Acquisition of Verb Hypothesis Space towards Physical World Interaction

Lanbo She and **Joyce Y. Chai**
Department of Computer Science and Engineering
Michigan State University
East Lansing, Michigan 48824, USA
{shelanbo, jchai}@cse.msu.edu

Abstract

As a new generation of cognitive robots start to enter our lives, it is important to enable robots to follow human commands and to learn new actions from human language instructions. To address this issue, this paper presents an approach that explicitly represents verb semantics through hypothesis spaces of fluents and automatically acquires these hypothesis spaces by interacting with humans. The learned hypothesis spaces can be used to automatically plan for lower-level primitive actions towards physical world interaction. Our empirical results have shown that the representation of a hypothesis space of fluents, combined with the learned hypothesis selection algorithm, outperforms a previous baseline. In addition, our approach applies incremental learning, which can contribute to life-long learning from humans in the future.

1 Introduction

As a new generation of cognitive robots start to enter our lives, it is important to enable robots to follow human commands (Tellex et al., 2014; Thomason et al., 2015) and to learn new actions from human language instructions (Cantrell et al., 2012; Mohan et al., 2013). To achieve such a capability, one of the fundamental challenges is to link higher-level concepts expressed by human language to lower-level primitive actions the robot is familiar with. While grounding language to perception (Gorniak and Roy, 2007; Chen and Mooney, 2011; Kim and Mooney, 2012; Artzi and Zettlemoyer, 2013; Tellex et al., 2014; Liu et al., 2014; Liu and Chai, 2015) has received much attention in recent years, less work has addressed grounding language to robotic action. Actions are often expressed by verbs or verb phrases. Most semantic representations for verbs are based on argument frames (e.g., thematic roles which capture participants of an action). For example, suppose a human directs a robot to "*fill the cup with milk*". The robot will need to first create a semantic representation for the verb "*fill*" where "*the cup*" and "*milk*" are grounded to the respective objects in the environment (Yang et al., 2016). Suppose the robot is successful in this first step, it still may not be able to execute the action "*fill*" as it does not know how this higher-level action corresponds to its lower-level primitive actions.

In robotic systems, operations usually consist of multiple segments of lower-level primitive actions (e.g., *move to*, *open gripper*, and *close gripper*) which are executed both sequentially and concurrently. Task scheduling provides the order or schedule for executions of different segments of actions and action planning provides the plan for executing each individual segment. Primitive actions are often predefined in terms of how they change the state of the physical world. Given a goal, task scheduling and action planning will derive a sequence of primitive actions that can change the initial environment to the goal state. The goal state of the physical world becomes a driving force for robot actions. Thus, beyond semantic frames, modeling verb semantics through their effects on the state of the world may provide a link to connect higher-level language and lower-level primitive actions.

Motivated by this perspective, we have developed an approach where each verb is explicitly represented by a hypothesis space of fluents (i.e., desired goal states) of the physical world, which is incrementally acquired and updated through interacting with humans. More specifically, given a human command, if there is no knowledge about the

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 108–117,
Berlin, Germany, August 7-12, 2016.

corresponding verb (i.e., no existing hypothesis space for that verb), the robot will initiate a learning process by asking human partners to demonstrate the sequence of actions that is necessary to accomplish this command. Based on this demonstration, a hypothesis space of fluents for that verb frame will be automatically acquired. If there is an existing hypothesis space for the verb, the robot will select the best hypothesis that is most relevant to the current situation and plan for the sequence of lower-level actions. Based on the outcome of the actions (e.g., whether it has successfully executed the command), the corresponding hypothesis space will be updated. Through this fashion, a hypothesis space for each encountered verb frame is incrementally acquired and updated through continuous interactions with human partners. In this paper, to focus our effort on representations and learning algorithms, we adopted an existing benchmark dataset (Misra et al., 2015) to simulate the incremental learning process and interaction with humans.

Compared to previous works (She et al., 2014b; Misra et al., 2015), our approach has three unique characteristics. First, rather than a single goal state associated with a verb, our approach captures a space of hypotheses which can potentially account for a wider range of novel situations when the verb is applied. Second, given a new situation, our approach can automatically identify the best hypothesis that fits the current situation and plan for lower-level actions accordingly. Third, through incremental learning and acquisition, our approach has a potential to contribute to life-long learning from humans. This paper provides details on the hypothesis space representation, the induction and inference algorithms, as well as experiments and evaluation results.

2 Related Work

Our work here is motivated by previous linguistic studies on verbs, action modeling in AI, and recent advances in grounding language to actions.

Previous linguistic studies (Hovav and Levin, 2008; Hovav and Levin, 2010) propose action verbs can be divided into two types: ***manner verbs*** that "specify as part of their meaning a manner of carrying out an action" (e.g., *nibble, rub, laugh, run, swim*), and ***result verbs*** that "specify the coming about of a result state" (e.g., *clean, cover, empty, fill, chop, cut, open, enter*). Recent work has shown that explicitly modeling resulting change of state for action verbs can improve grounded language understanding (Gao et al., 2016). Motivated by these studies, this paper focuses on result verbs and uses hypothesis spaces to explicitly represent the result states associated with these verbs.

In AI literature on action modeling, action schemas are defined with preconditions and effects. Thus, representing verb semantics for action verbs using resulting states can be connected to the agent's underlying planning modules. Different from earlier works in the planning community that learn action models from example plans (Wang, 1995; Yang et al., 2007) and from interactions (Gil, 1994), our goal here is to explore the representation of verb semantics and its acquisition through language and action.

There has been some work in the robotics community to translate natural language to robotic operations (Kress-Gazit et al., 2007; Jia et al., 2014; Sung et al., 2014; Spangenberg and Henrich, 2015), but not for the purpose of learning new actions. To support action learning, previously we have developed a system where the robot can acquire the meaning of a new verb (e.g., *stack*) by following human's step-by-step language instructions (She et al., 2014a; She et al., 2014b). By performing the actions at each step, the robot is able to acquire the desired goal state associated with the new verb. Our empirical results have shown that representing acquired verbs by resulting states allow the robot to plan for primitive actions in novel situations. Moreover, recent work (Misra et al., 2014; Misra et al., 2015) has presented an algorithm for grounding higher-level commands such as "microwave the cup" to lower-level robot operations, where each verb lexicon is represented as the desired resulting states. Their empirical evaluations once again have shown the advantage of representing verbs as desired states in robotic systems. Different from these previous works, we represent verb semantics through a hypothesis space of fluents (rather than a single hypothesis). In addition, we present an incremental learning approach for inducing the hypothesis space and selecting the best hypothesis.

3 An Incremental Learning Framework

An overview of our incremental learning framework is shown in Figure 1. Given a language

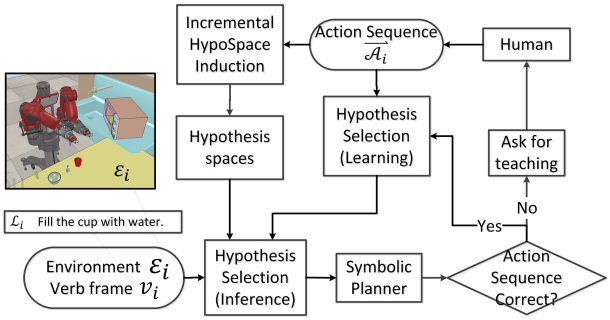


Figure 1: An incremental process of verb acquisition (i.e. learning) and application (i.e. inference).

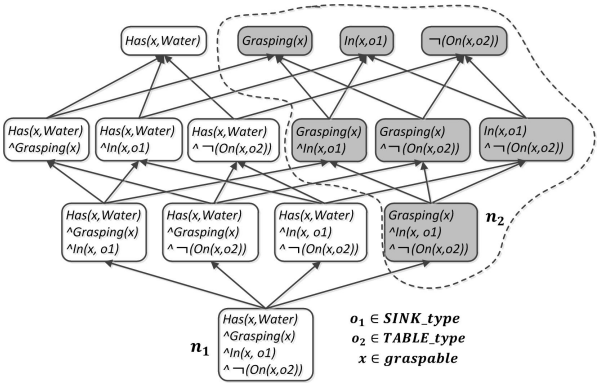


Figure 2: An example hypothesis space for the verb frame `fill(x)`. The bottom node captures the state changes after executing the *fill* command in the environment. Anchored by the bottom node, the hypothesis space is generated in a bottom-up fashion. Each node represents a potential goal state. The highlighted nodes are pruned during induction, as they are not consistent with the bottom node.

command $\mathcal{L}_i$ (e.g. *"fill the cup with water."*) and an environment $\mathcal{E}_i$ (e.g. a simulated environment shown in Figure 1), the goal is to identify a sequence of lower-level robotic actions to perform the command. Similar to previous works (Pasula et al., 2007; Mouro et al., 2012), the environment $\mathcal{E}_i$ is represented by a conjunction of grounded state fluents, where each fluent describes either the property of an object or relations (e.g. spatial) between objects. The language command $\mathcal{L}_i$ is first translated to an intermediate representation of grounded verb frame v_i through semantic parsing and referential grounding (e.g. for "*fill the cup*", the argument *the cup* is grounded to `Cup1` in the scene). The system knowledge of each verb frame (e.g., `fill(x)`) is represented by a *Hypothesis Space* $\mathcal{H}$, where each hypothesis (i.e. a node) is a description of possible fluents - or, in other words, resulting states - that are attributed to executing the verb command. Given a verb frame v_i and an environment $\mathcal{E}_i$, a *Hypothesis Selector* will choose an optimal hypothesis from space $\mathcal{H}$ to describe the expected resulting state of executing v_i in $\mathcal{E}_i$. Given this goal state and the current environment, a symbolic planner such as the STRIPS planner (Fikes and Nilsson, 1971) is used to generate an action sequence for the agent to execute. If the action sequence correctly performs the command (e.g. as evaluated by a human partner), the hypothesis selector will be updated with the success of its prediction. On the other hand, if the action has never been encountered (i.e., the system has no knowledge about this verb and thus the corresponding space is empty) or the predicted action sequence is incorrect, the human partner will provide an action sequence $\vec{A}_i$ that can correctly perform command v_i in the current environment. Using $\vec{A}_i$ as the ground truth information, the system will not only update the hypothesis selector, but will also update the existing space of v_i. The updated hypothesis space is treated as system knowledge of v_i, which will be used in future interaction. Through this procedure, a hypothesis space for each verb frame v_i is continually and incrementally updated through human-robot interaction.

4 State Hypothesis Space

To bridge human language and robotic actions, previous works have studied representing the semantics of a verb with a single resulting state (She et al., 2014b; Misra et al., 2015). One problem of this representation is that when the verb is applied in a new situation, if any part of the resulting state cannot be satisfied, the symbolic planner will not be able to generate a plan for lower-level actions to execute this verb command. The planner is also not able to determine whether the failed part of state representation is even necessary. In fact, this effect is similar to the over-fitting problem. For example, given a sequence of actions of performing `fill(x)`, the induced hypothesis could be "$Has(x, Water) \wedge Grasping(x) \wedge In(x, o_1) \wedge \neg(On(x, o_2))$", where x is a graspable object (e.g. a cup or bowl), o_1 is any type of sink, and o_2 is any table. However, during inference, when applied to a new situation that does not have any type of sink or table, this hypothesis will not

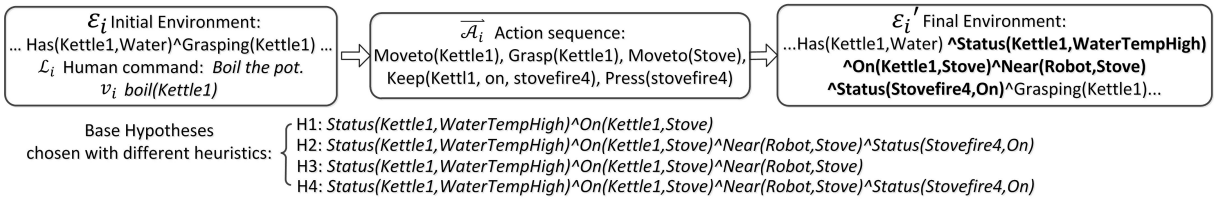


Figure 3: A training instance $\{\mathcal{E}_i, v_i, \vec{\mathcal{A}}_i\}$ for hypothesis space induction. $\mathcal{E}_i'$ is the resulting environment of executing $\vec{\mathcal{A}}_i$ in $\mathcal{E}_i$. The change of state in $\mathcal{E}_i'$ compared to $\mathcal{E}_i$ is highlighted in bold. Different heuristics generate different Base Hypotheses as shown at the bottom.

be applicable. Nevertheless, the first two terms $Has(x, Water) \wedge Grasping(x)$ may already be sufficient to generate a plan for completing the verb command.

To handle this over-fitting problem, we propose a hierarchical hypothesis space to represent verb semantics, as shown in Figure 2. The space is organized based on a specific-to-general hierarchical structure. Formally, a hypothesis space $\mathcal{H}$ for a verb frame is defined as: $\langle \mathbb{N}, \mathbb{E} \rangle$, where each $n_i \in \mathbb{N}$ is a hypothesis node and each $e_{ij} \in \mathbb{E}$ is a directed edge pointing from parent n_i to child n_j, meaning node n_j is more general than n_i and has one less constraint.

In Figure 2, the bottom hypothesis (n_1) is $Has(x, Water) \wedge Grasping(x) \wedge In(x, o1) \wedge \neg(On(x, o2))$. A hypothesis n_i represents a conjunction of parameterized state fluents l_k:

$$n_i := \wedge l_k, and\, l_k := [\neg]\, pred_k(x_{k_1}[, x_{k_2}])$$

A fluent l_k is composed of a predicate (e.g. object status: *Has,* or spatial relation: *On*) and a set of argument variables. It can be positive or negative. Take the bottom node in Figure 2 as an example, it contains four fluents including one negative term (i.e. $\neg(On(x, o_2))$) and three positive terms. During inference, the parameters will be grounded to the environment to check whether this hypothesis is applicable.

5 Hypothesis Space Induction

Given an initial environment $\mathcal{E}_i$, a language command which contains the verb frame v_i, and a corresponding action sequence $\vec{\mathcal{A}}_i$, $\{\mathcal{E}_i, v_i, \vec{\mathcal{A}}_i\}$ forms a training instance for hypothesis space induction. First, based on different heuristics, a base hypothesis is generated by comparing the state difference between the final and the initial environment. Second, a hypothesis space $\mathcal{H}$ is induced on top of this *Base Hypothesis* in a bottom-up fashion. And during induction some nodes are pruned. Third, if the system has existing knowledge for the same verb frame (i.e. an existing hypothesis space $\mathcal{H}_t$ for the same verb frame), this newly induced space will be merged with previous knowledge. Next we explain each step in detail.

5.1 Base Hypothesis Induction

One key concept in the space induction is the *Base Hypothesis* (e.g. the bottom node in Figure 2), which provides a foundation for building a space. As shown in Figure 3, given a verb frame v_i and a working environment $\mathcal{E}_i$, the action sequence $\vec{\mathcal{A}}_i$ given by a human will change the initial environment $\mathcal{E}_i$ to a final environment $\mathcal{E}_i'$. The state changes are highlighted in Figure 3. Suppose a state change can be described by n fluents. Then the first question is which of these n fluents should be included in the base hypothesis. To gain some understanding on what would be a good representation, we applied different heuristics of choosing fluents to form a base hypothesis as shown in Figure 3:

- $H1_{argonly}$: only includes the changed states associated with the argument objects specified in the frame (e.g., in Figure 3, `Kettle1` is the only argument).
- $H2_{manip}$: includes the changed states of all the objects that have been manipulated in the action sequence taught by the human.
- $H3_{argrelated}$: includes the changed states of all the objects related to the argument objects in the final environment. An object o is considered as "related to" an argument object if there is a state fluent that includes both o and an argument object in one predicate. (e.g. `Stove` is related to the argument object `Kettle1` through $On(Kettle1, Stove)$).

Input: A Base Hypothesis h
Initialization: Set initial space $\mathcal{H}$: $\langle \mathbb{N}, \mathbb{E} \rangle$ with $\mathbb{N}$:[h] and $\mathbb{E}$:[ ],
Set a set of temporary hypotheses T:[h]
while T is not empty **do**
  Pop an element t from T
  Generate children [$t^{(0)}$,...,$t^{(k)}$] from t by removing each single fluent
  foreach i = 0 ... k **do**
    if $t^{(i)}$ is consistent with t **then**
      Append $t^{(i)}$ to T;
      Add $t^{(i)}$ to $\mathbb{N}$ if not already in;
      Add link $t \rightarrow t^{(i)}$ to $\mathbb{E}$ if not already in;
    else
      Prune $t^{(i)}$ and any node that can be generalized from $t^{(i)}$
    end
  end
end
Output: Hypothesis space $\mathcal{H}$

Algorithm 1: A single hypothesis space induction algorithm. $\mathcal{H}$ is a space initialized with a base hypothesis and an empty set of links. T is a temporary container of candidate hypotheses.

- $H4_{all}$: includes all the fluents whose values are changed from $\mathcal{E}_i$ to $\mathcal{E}_i'$ (e.g. all the four highlighted state fluents in $\mathcal{E}_i'$).

5.2 Single Space Induction

First we define the ***consistency*** between two hypotheses:

Definition. Hypotheses h_1 and h_2 are ***consistent***, if and only if the action sequence $\vec{\mathcal{A}}_1$ generated from a symbolic planner based on goal state h_1 is exactly the same as the action sequence $\vec{\mathcal{A}}_2$ generated based on goal state h_2.

Given a base hypothesis, the space induction process is a while-loop generalizing hypotheses in a bottom-up fashion, which stops when no hypotheses can be further generalized. As shown in Algorithm 1, a hypothesis node t can firstly be generalized to a set of immediate children [$t^{(0)}$,...,$t^{(k)}$] by removing a single fluent from t. For example, the base hypothesis n_1 in Figure 2 is composed of 4 fluents, such that 4 immediate children nodes can potentially be generated. If a child node $t^{(i)}$ is consistent with its parent t (i.e. determined based on the *consistency* defined previously), node $t^{(i)}$ and a link $t \rightarrow t^{(i)}$ are added to the space $\mathcal{H}$. The node $t^{(i)}$ is also added to a temporary hypothesis container waiting to be further generalized. On the other hand, some children hypotheses can be inconsistent with their parents. For example, the gray node (n_2) in Figure 2 is a child node that is inconsistent with its parent (n_1). As n_2 does not explicitly specify $Has(x, Water)$ as part of its goal state, the symbolic planner generates less steps to achieve goal state n_2 than goal state n_1. This implies that the semantics of achieving n_2 may be different than those for achieving n_1. Such hypotheses that are inconsistent with their parents are pruned. In addition, if $t^{(i)}$ is inconsistent with its parent t, any children of $t^{(i)}$ are also inconsistent with t (e.g. children of n_2 in Figure 2 are also gray nodes, meaning they are inconsistent with the base hypothesis). Through pruning, the size of entire space can be greatly reduced.

In the resulting hypothesis space, every single hypothesis is consistent with the base hypothesis. By only keeping consistent hypotheses via pruning, we can remove fluents that are not representative of the main goal associated with the verb.

5.3 Space Merging

If the robot has existing knowledge (i.e. hypothesis space $\mathcal{H}_t$) for a verb frame, the induced hypothesis space $\mathcal{H}$ from a new instance of the same verb will be merged with the existing space $\mathcal{H}_t$. Currently, a new space $\mathcal{H}_{t+1}$ is generated where the nodes of $\mathcal{H}_{t+1}$ are the union of $\mathcal{H}$ and $\mathcal{H}_t$, and links in $\mathcal{H}_{t+1}$ are generated by checking the parent-child relationship between nodes. In future work, more space merging operations will be explored, and human feedback will be incorporated into the induction process.

6 Hypothesis Selection

Hypothesis selection is applied when the agent intends to execute a command. Given a verb frame extracted from the language command, the agent will first select the best hypothesis (describing the goal state) from the existing knowledge base, and then apply a symbolic planner to generate an action sequence to achieve the goal. In our framework, the model of selecting the best hypothesis is incrementally learned throughout continuous interaction with humans. More specifically, given a correct action sequence (whether performed by the robot or provided by the human), a regression model is trained to capture the fitness of a hypothesis given a particular situation.

Inference: Given a verb frame v_i and a working environment $\mathcal{E}_i$, the goal of inference is to estimate how well each hypothesis h_k from a space $\mathcal{H}_t$ describes the expected result of performing v_i

in $\mathcal{E}_i$. The best fit hypothesis will be used as the goal state to generate the action sequence. Specifically, the "goodness" of describing command v_i with hypothesis h_k in environment $\mathcal{E}_i$ is formulated as follows:

$$f(h_k \mid v_i; \mathcal{E}_i; \mathcal{H}_t) = W^T \cdot \Phi(h_k, v_i, \mathcal{E}_i, \mathcal{H}_t) \quad (1)$$

where $\Phi(h_k, v_i, \mathcal{E}_i, \mathcal{H}_t)$ is a feature vector capturing multiple aspects of relations between $h_k, v_i, \mathcal{E}_i$ and $\mathcal{H}_t$ as shown in Table 1; and W captures the weight associated with each feature. Example global features include whether the candidate goal h_k is in the top level of entire space $\mathcal{H}_t$ and whether h_k has the highest frequency. Example local features include if most of the fluents in h_k are already satisfied in current scene $\mathcal{E}_i$ (as this h_k is unlikely to be a desired goal state). The features also include whether the same verb frame v_i has been performed in a similar scene during previous interactions, as the corresponding hypotheses induced during that experience are more likely to be relevant and are thus preferred.

Parameter Estimation: Given an action sequence $\vec{\mathcal{A}}_i$ that illustrates how to correctly perform command v_i in environment $\mathcal{E}_i$ during interaction, the model weights will be incrementally updated with[1]:

$$W_{t+1} = W_t - \eta(\alpha \frac{\partial R(W_t)}{\partial W_t} + \frac{\partial L(J_{ki}, f_{ki})}{\partial W_t})$$

where $f_{ki} := f(h_k|v_i; \mathcal{E}_i; \mathcal{H}_t)$ is defined in Equation 1. J_{ki} is the dependent variable the model should approximate, where $J_{ki} := J(s_i, h_k)$ is the Jaccard Index (details in Section 7) between hypothesis h_k and a set of changed states s_i (i.e. the changed states of executing the illustration action sequence $\vec{\mathcal{A}}_i$ in current environment). $L(J_{ki}, f_{ki})$ is a squared loss function. $\alpha R(W_t)$ is the penalty term, and η is the constant learning rate.

7 Experiment Setup

Dataset Description. To evaluate our approach, we applied the dataset made available by (Misra et al., 2015). To support incremental learning, each utterance from every original paragraph is extracted so that each command/utterance only contains one verb and its arguments. The corresponding initial environment and an action sequence taught by a human for each command are also extracted. An example is shown in Figure 3, where $\mathcal{L}_i$ is a language command, $\mathcal{E}_i$ is the initial working environment, and $\vec{\mathcal{A}}_i$ is a sequence of primitive actions to complete the command given by the human. In the original data, some sentences are not aligned with any actions, and thus cannot be used for either the learning or the evaluation. Removing these unaligned sentences resulted in a total of 991 data instances, including 165 different verb frames.

Features on candidate hypothesis h_k and the space $\mathcal{H}_t$ 1. If h_k belongs to the top level of $\mathcal{H}_t$. 2. If h_k has the highest frequency in $\mathcal{H}_t$.
Features on h_k and current situation $\mathcal{E}_i$ 3. Portion of fluents in h_k that are already satisfied by $\mathcal{E}_i$. 4. Portion of non-argument objects in h_k. Examples of non-argument objects are o_1 and o_2 in Figure 2.
Features on relations between a testing verb frame v_i and previous interaction experience 5. Whether the same verb frame v_i has been executed previously with the same argument objects. 6. Similarities between noun phrase descriptions used in current command and commands from interaction history.

Table 1: Current features used for incremental learning of the regression model. The first two are binary features and the rest are real-valued features.

Among the 991 data instances, 793 were used for incremental learning (i.e., space induction and hypothesis selector learning). Specifically, given a command, if the robot correctly predicts an action sequence[2], this correct prediction is used to update the hypothesis selector. Otherwise, the agent will require a correct action sequence from the human, which is used for hypothesis space induction as well as updating the hypothesis selector.

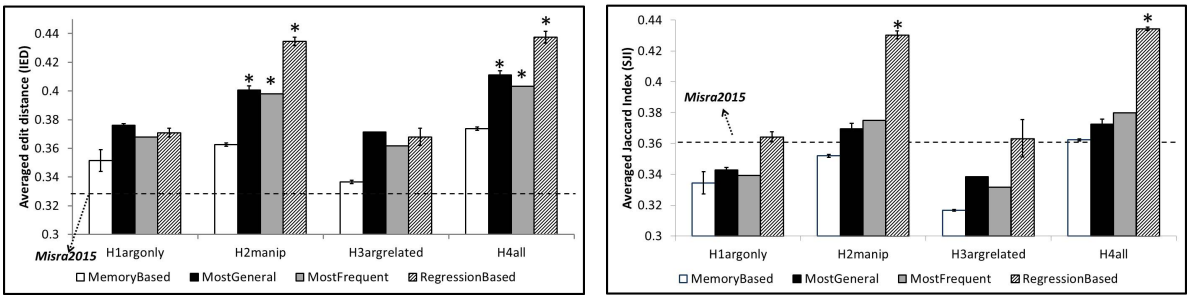
The hypothesis spaces and regression based selectors acquired at each run were evaluated on the other 20% (198) testing instances. Specifically, for each testing instance, the induced space and the hypothesis selector were applied to identify a desired goal state. Then a symbolic planner[3] was applied to predict an action sequence $\vec{\mathcal{A}}^{(p)}$ based on this predicted goal state. We then compared $\vec{\mathcal{A}}^{(p)}$ with the ground truth action sequence $\vec{\mathcal{A}}^{(g)}$ using the following two metrics.

- IED (*Instruction Edit Distance*) measures

[1]The SGD regressor in the scikit-learn (Pedregosa et al., 2011) is used to perform the linear regression with L2 regularization.

[2]Currently, a prediction is considered correct if the predicted result ($c^{(p)}$) is similar to a human labeled action sequence ($c^{(g)}$) (i.e., $SJI(c^{(g)}, c^{(p)}) > 0.5$).

[3]The symbolic planner implemented by (Rintanen, 2012) was utilized to generate action sequences.



(a) IED results for different configurations

(b) SJI results for different configurations

Figure 4: The overall performance on the testing set with different configurations in generating the base hypothesis and in hypothesis selection. Each configuration runs five times by randomly shuffling the order of learning instances, and the averaged performance is reported. The result from *Misra2015* is shown as a line. Results that are statistically significant better than *Misra2015* are marked with $*$ (*paired t-test,* $p< 0.05$).

similarity between the ground truth action sequence $\vec{\mathcal{A}}^{(g)}$ and the predicted sequence $\vec{\mathcal{A}}^{(p)}$. Specifically, the edit distance d between two action sequences $\vec{\mathcal{A}}^{(g)}$ and $\vec{\mathcal{A}}^{(p)}$ is first calculated. Then d is rescaled as $IED = 1 - d/max(\vec{\mathcal{A}}^{(g)}, \vec{\mathcal{A}}^{(p)})$, such that IED ranges from 0 to 1 and a larger IED means the two sequences are more similar.

- SJI (*State Jaccard Index*). Because different action sequences could lead to a same goal state, we also use Jaccard Index to check the overlap between the changed states. Specifically, executing the ground truth action sequence $\vec{\mathcal{A}}^{(g)}$ in the initial scene $\mathcal{E}_i$ results in a final environment $\mathcal{E}_i'$. Suppose the changed states between $\mathcal{E}_i$ and $\mathcal{E}_i'$ is $c^{(g)}$. For the predicted action sequence, we can calculate another set of changed states $c^{(p)}$. The Jaccard Index between $c^{(g)}$ and $c^{(p)}$ is evaluated, which also ranges from 0 to 1 and a larger SJI means the predicted state changes are more similar to the ground truth.

Configurations. We also compared the results of using the regression based selector to select a hypothesis (i.e., *RegressionBased*) with the following different strategies for selecting the hypothesis:

- *Misra2015*: The state of the art system reported in (Misra et al., 2015) on the command/utterance level evaluation[4].

- *MemoryBased*: Given the induced space, only the base hypotheses h_ks from each learning instances are used. Because these h_ks don't have any relaxation, they represent purely learning from memorization.

- *MostGeneral*: In this case, only those hypotheses from the top level of the hypothesis space are used, which contain the least number of fluents. These nodes are the most relaxed hypotheses in the space.

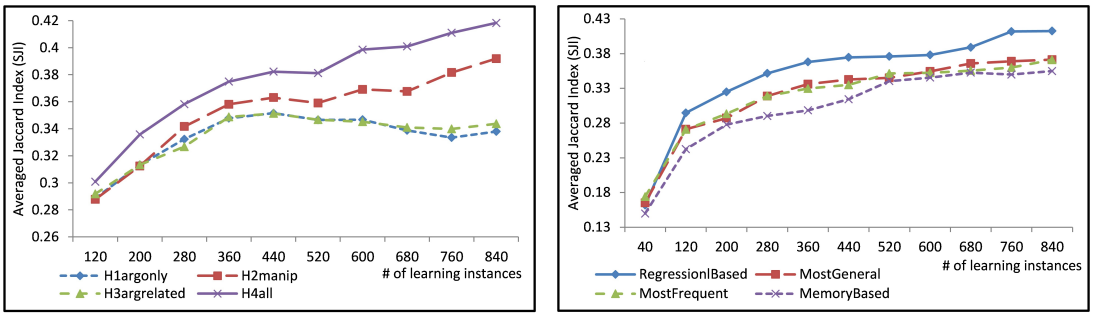
- *MostFrequent*: In this setting, the hypotheses that are most frequently observed in the learning instances are used.

8 Results

8.1 Overall performance

The results of the overall performance across different configurations are shown in Figure 4. For both of the IED and SJI (i.e. Figure 4(a) and Figure 4(b)), the hypothesis spaces with the regression model based hypothesis selector always achieve the best performance across different configurations, and outperforms the previous approach (Misra et al., 2015). For different base hypothesis induction strategies, the $H4_{all}$ considering all the changed states achieves the best performance across all configurations. This is because $H4_{all}$ keeps all of the state change information compared with other heuristics. The performance of $H2_{manip}$ is similar to $H4_{all}$. The reason is that, when all the manipulated objects are considered, the resulted set of changed states will cover most of the fluents in $H4_{all}$. On the other dimension,

[4]We applied the same system described in (Misra et al., 2015) to predict action sequences. The only difference is here we report the performance at the command level, not at the paragraph level.



(a) Use regression based selector to select hypothesis, and compare each base hypothesis induction heuristics.

(b) Induce the base hypothesis with $H4_{all}$, and compare different hypothesis selection strategies.

Figure 5: Incremental learning results. The spaces and regression models acquired at different incremental learning cycles are evaluated on testing set. The averaged Jaccard Index is reported.

the regression based hypothesis selector achieves the best performance and the *MemoryBased* strategy has the lowest performance. Results for *MostGeneral* and *MostFrequent* are between the regression based selector and *MemoryBased*.

8.2 Incremental Learning Results

Figure 5 presents the incremental learning results on the testing set. To better present the results, we show the performance based on each learning cycle of 40 instances. The averaged Jaccard Index (SJI) is reported. Specifically, Figure 5(a) shows the results of configurations comparing different base hypothesis induction heuristics using regression model based hypothesis selection. After using 200 out of 840 (23.8%) learning instances, all the four curves achieve more than 80% of the overall performance. For example, for the heuristic $H4_{all}$, the final average Jaccard Index is 0.418. When 200 instances are used, the score is 0.340 ($0.340/0.418 \approx 81\%$). The same number holds for the other heuristics. After 200 instances, $H4_{all}$ and $H2_{manip}$ consistently achieve better performance than $H1_{argonly}$ and $H3_{argrelated}$. This result indicates that while change of states mostly affect the arguments of the verbs, other state changes in the environment cannot be ignored. Modeling them actually leads to better performance. Using $H4_{all}$ for base hypothesis induction, Figure 5(b) shows the results of comparing different hypothesis selection strategies. The regression model based selector always outperforms other selection strategies.

8.3 Results on Frequently Used Verb Frames

Beside overall evaluation, we have also taken a closer look at individual verb frames. Most of the verb frames in the data have a very low frequency, which cannot produce statistically significant results. So we only selected verb frames with frequency larger than 40 in this evaluation. For each verb frame, 60% data are used for incremental learning and 40% are for testing. For each frame, a regression based selector is trained separately. The resulting SJI curves are shown in Figure 6.

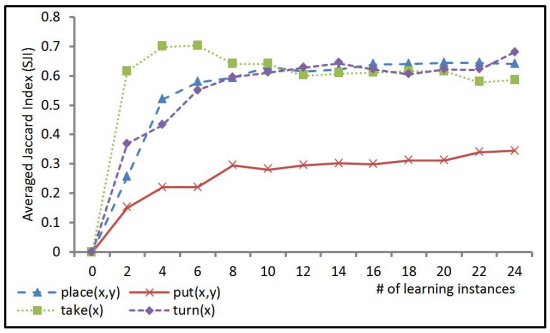


Figure 6: Incremental evaluation for individual verb frames. Four frequently used verb frames are examined: $place(x, y)$, $put(x, y)$, $take(x)$, and $turn(x)$. X-axis is the number of incremental learning instances, and Y-axis is the averaged SJI computed with $H4_{all}$ base hypothesis induction and regression based hypothesis selector.

As shown in Figure 6, all the four curves become steady after 8 learning instances are used. However, while some verb frames have final SJIs of more than 0.55 (i.e. $take(x)$ and $turn(x)$), others have relatively lower results (e.g. results for $put(x, y)$ are lower than 0.4). After examining the learning instances for $put(x, y)$, we found these data are more noisy than the training data for other frames. One source of errors is the incorrect object grounding results. For example, a problematic

training instance is "*put the pillow on the couch*", where the object grounding module cannot correctly ground the "*couch*" to the target object. As a result, the changed states of the second argument (i.e. the "*couch*") are incorrectly identified, which leads to incorrect prediction of desired states during inference. Another common error source is from automated parsing of utterances. The action frames generated from the parsing results could be incorrect in the first place, which would contribute to a hypothesis space for a wrong frame. These different types of errors are difficult to be recognized by the system itself. This points to the future direction of involving humans in a dialogue to learn a more reliable hypothesis space for verb semantics.

9 Conclusion

This paper presents an incremental learning approach that represents and acquires semantics of action verbs based on state changes of the environment. Specifically, we propose a hierarchical hypothesis space, where each node in the space describes a possible effect on the world from the verb. Given a language command, the induced hypothesis space, together with a learned hypothesis selector, can be applied by the agent to plan for lower-level actions. Our empirical results have demonstrated a significant improvement in performance compared to a previous leading approach. More importantly, as our approach is based on incremental learning, it can be potentially integrated in a dialogue system to support life-long learning from humans. Our future work will extend the current approach with dialogue modeling to learn more reliable hypothesis spaces of resulting states for verb semantics.

Acknowledgments

This work was supported by IIS-1208390 and IIS-1617682 from the National Science Foundation. The authors would like to thank Dipendra K. Misra and colleagues for providing the evaluation data, and the anonymous reviewers for valuable comments.

References

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, Volume1(1):49–62.

R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz. 2012. Tell me when and why to do it! run-time planner model updates via natural language instruction. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI'12)*, pages 471–478, Boston, Massachusetts, USA, March.

David L Chen and Raymond J Mooney. 2011. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-2011)*, pages 859–865, San Francisco, California, USA, August.

Richard E. Fikes and Nils J. Nilsson. 1971. Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI'71)*, pages 608–620, London, England.

Qiaozi Gao, Malcolm Doering, Shaohua Yang, and Joyce Y. Chai. 2016. Physical causality of action verbs in grounded language understanding. In *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany.

Yolanda Gil. 1994. Learning by experimentation: incremental refinement of incomplete planning domains. In *Prococeedings of the Eleventh International Conference on Machine Learning (ICML'94)*, pages 87–95, New Brunswick, NJ, USA.

P. Gorniak and D. Roy. 2007. Situated language understanding as filtering perceived affordances. *Cognitive Science*, Volume31(2):197–231.

Malka Rappaport Hovav and Beth Levin. 2008. Reflections on manner/result complementarity. *Lecture notes*.

Malka Rappaport Hovav and Beth Levin. 2010. Reflections on Manner / Result Complementarity. *Lexical Semantics, Syntax, and Event Structure*, pages 21–38.

Yunyi Jia, Ning Xi, Joyce Y. Chai, Yu Cheng, Rui Fang, and Lanbo She. 2014. Perceptive feedback for natural language control of robotic operations. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 6673–6678.

Joohyun Kim and Raymond J. Mooney. 2012. Unsupervised pcfg induction for grounded language learning with highly ambiguous supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and Natural Language Learning (EMNLP-CoNLL '12)*, pages 433–444, Jeju Island, Korea.

Hadas Kress-Gazit, Georgios E Fainekos, and George J Pappas. 2007. From structured english to robot motion. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2717–2722.

Changsong Liu and Joyce Y. Chai. 2015. Learning to mediate perceptual differences in situated human-robot dialogue. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 2288–2294, Austin, Texas, USA.

Changsong Liu, Lanbo She, Rui Fang, and Joyce Y. Chai. 2014. Probabilistic labeling for efficient referential grounding based on collaborative discourse. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics ACL'14 (Volume 2: Short Papers)*, pages 13–18, Baltimore, MD, USA.

Dipendra Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. 2014. Tell me dave: Context-sensitive grounding of natural language to mobile manipulation instructions. In *Proceedings of Robotics: Science and Systems (RSS'14)*, Berkeley, US.

Dipendra Kumar Misra, Kejia Tao, Percy Liang, and Ashutosh Saxena. 2015. Environment-driven lexicon induction for high-level instructions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing ACL-IJCNLP'15 (Volume 1: Long Papers)*, pages 992–1002, Beijing, China.

Shiwali Mohan, James Kirk, and John Laird. 2013. A computational model for situated task learning with interactive instruction. In *Proceedings of the International conference on cognitive modeling (ICCM'13)*.

Kira Mouro, Luke S. Zettlemoyer, Ronald P. A. Petrick, and Mark Steedman. 2012. Learning strips operators from noisy and incomplete observations. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence (UAI'12)*, pages 614–623, Catalina Island, CA, USA.

Hanna M Pasula, Luke S Zettlemoyer, and Leslie Pack Kaelbling. 2007. Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, Volume29:309–352.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, Volume12:2825–2830.

Jussi Rintanen. 2012. Planning as satisfiability: Heuristics. *Artificial Intelligence*, Volume193:45–86.

Lanbo She, Yu Cheng, Joyce Y. Chai, Yunyi Jia, Shaohua Yang, and Ning Xi. 2014a. Teaching robots new actions through natural language instructions. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication, IEEE RO-MAN'14*, pages 868–873, Edinburgh, UK.

Lanbo She, Shaohua Yang, Yu Cheng, Yunyi Jia, Joyce Y. Chai, and Ning Xi. 2014b. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 89–97, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

M. Spangenberg and D. Henrich. 2015. Grounding of actions based on verbalized physical effects and manipulation primitives. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 844–851, Hamburg, Germany.

Jaeyong Sung, Bart Selman, and Ashutosh Saxena. 2014. Synthesizing manipulation sequences for under-specified tasks using unrolled markov random fields. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14)*, pages 2970–2977, Chicago, IL, USA.

Stefanie Tellex, Pratiksha Thaker, Joshua Joseph, and Nicholas Roy. 2014. Learning perceptually grounded word meanings from unaligned parallel data. *Machine Learning*, Volume94(2):151–167.

Jesse Thomason, Shiqi Zhang, Raymond Mooney, and Peter Stone. 2015. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the 2015 International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1923–1929, Buenos Aires, Argentina.

Xuemei Wang. 1995. Learning by observation and practice: An incremental approach for planning operator acquisition. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML'95)*, pages 549–557, Tahoe City, California, USA.

Qiang Yang, Kangheng Wu, and Yunfei Jiang. 2007. Learning action models from plan examples using weighted max-sat. *Artificial Intelligence*, Volume171(23):107 – 143.

Shaohua Yang, Qiaozi Gao, Changsong Liu, Caiming Xiong, Song-Chun Zhu, and Joyce Y. Chai. 2016. Grounded semantic role labeling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*, San Diego, California.