Learning linguistic constructions grounded in qualitative action models

Maximilian Panzner, Judith Gaspers and Philipp Cimiano

Abstract— **Aiming at the design of adaptive artificial agents which are able to learn autonomously from experience and human tutoring, in this paper we present a system for learning syntactic constructions grounded in perception. These constructions are learned from examples of natural language utterances and parallel performances of actions, i.e. their trajectories and involved objects. From the input, the system learns linguistic structures and qualitative action models. Action models are represented as Hidden Markov Models over sequences of qualitative relations between a trajector and a landmark and abstract away from concrete action trajectories. Learning of action models is driven by linguistic observations, and linguistic patterns are, in turn, grounded in learned action models. The proposed system is applicable for both language understanding and language generation. We present empirical results, showing that the learned action models generalize well over concrete instances of the same action and also to novel performers, while allowing accurate discrimination between different actions. Further, we show that the system is able to describe novel dynamic scenes and to understand novel utterances describing such scenes.**

I. Introduction

An important goal in order to achieve more natural and intuitive communication with robots is to equip them with the ability to learn language – in particular linguistic structures grounded in perception – autonomously, both from experience and via human tutoring. A particular challenge lies in endowing robots with the ability for open-ended language and action learning over their whole lifetime, thus enabling them to acquire new linguistic constructions and the actions or objects they refer to incrementally and continuously. Relying on grammars or linguistic knowledge encoded at design time clearly does not fulfill this goal.

To equip robots with the ability of open-ended language and action learning, the robot must be able to i) learn representations for objects and actions appearing in visual input, ii) extract words and grammatical patterns from a speech stream and iii) establish connections between the extracted structures to ground linguistic structures in models of perceptual observations. While work concerning the individual subtasks exists, a unified system solving all of them does not yet exist. Working towards such a system, in this paper we explore how syntactic patterns grounded in qualitative action models can be learned autonomously based on example performances of actions coupled with natural language descriptions. The acquisition of grammatical patterns along with a mapping to corresponding semantic representations has been explored previously by several researches, both with the goal of building cognitive models of human language acquisition (e.g. [1], [2], [3]) and with respect to application on a robot (e.g. [4], [5], [6]). However, most research has considered meaning representations in symbolic form, e.g. in predicate logic, rather than information extracted from video or images. While in particular some research aiming to equip robots with language learning abilities has also considered information extracted from scene representations, work has mainly focused on the description of static scenes (e.g. [6]), e.g. grounding language in objects or object positions rather than representations of actions. However, in order to learn syntactic patterns referring to actions, for instance, to allow human tutoring of novel actions, the robot also needs to be able to build models for actions which abstract over individual trajectories and ground linguistic knowledge in these. While previous work has also addressed learning action models (e.g. [7]), we are not aware of other systems learning these together with syntactic constructions.

Our approach is based on an existing cognitive model [1] for the acquisition of syntactic constructions which learns from symbolic input, i.e. from natural language (*NL*) utterances coupled with meaning representations represented by formulas in predicate logic. Extending this model towards learning from dynamic scenes rather than symbolic meaning representations, in this paper we explore learning from trajectories of actions and present an approach which learns generalized action models from such observations. We integrate this approach into the model such that learning of action models is driven by linguistic observations and linguistic patterns are, in turn, grounded in learned action models. Actions are represented as Hidden Markov Models over sequences of qualitative relations between a trajector and a landmark. Since we focus on learning action models, we assume that the system has the capability to extract representations for objects, and thus observed objects involved in actions are given in symbolic form.

For evaluation, we collected a reference dataset in which 12 human subjects performed different actions described by corresponding utterances, thus yielding several concrete examples of utterances and trajectories of actions described by these. We present empirical results on the dataset, showing that the learned action models generalize well over concrete instances of the same action and also to novel performers, while allowing accurate discrimination between different actions. Further, we show that by learning syntactic constructions grounded in the learned models, the system

*This work has been funded by the DFG within the CRC 673 and the Cognitive Interaction Technology Excellence Center.

The authors are with the Semantic Computing Group, CITEC, Bielefeld University, Bielefeld, Germany mpanzner@cit-ec.uni-bielefeld.de

is able to describe novel dynamic scenes and to understand novel utterances corresponding to such scenes.
The remainder of this paper is as follows. Next, we describe related work and subsequently the proposed approach. We then describe the learning setting and acquisition of input data and present empirical results on the acquired dataset.

II. Related Work

Our work is related both to approaches to grounded acquisition of language in robots and cognitive systems, but also to approaches to the representation and acquisition of actions. With respect to approaches to grounded acquisition of language, there has been a lot of work on developing models which can acquire single words and their meanings (e.g. [8], [9]). In some approaches, this meaning is grounded in perception, but is typically limited to static objects [10], [11]. Other approaches (e.g. [4], [5]) deal with the acquisition of syntactic constructions as we do, but typically do not ground these constructions in qualitative action models. The work by Feldman et al. on Embodied Construction Grammar (ECG) [12], [13] is very related to our approach. However, the X-Schemas in which the meaning of linguistic constructions are represented are very symbolic compared to our qualitative models. Our qualitative models are still far away from a full grounding in the sensoric and actuator systems of a robot, but clearly go one step further than the X-Schemas used in Embodied Construction Grammar (ECG). The closest related work is the one of van Trijp et al. [14], who have developed approaches by which robots can learn linguistic knowledge in the framework of Fluid Construction Grammar (FCG). With respect to the representation and acquisition of actions, different approaches based on prototypes [15], markov models [7] and neural networks [16] have been proposed. Suguria, Iwahashi, Kashioka and Nakamura [7] learn reference point dependent motion between a trajector and a set of possible landmarks as a quantitative sequence. They directly use position, velocity and acceleration relative to an action intrinsic coordinate system to learn a Hidden Markov Model for a given set of trajectories. Ogawara and Takamatsu [17] cluster independent trajectories using a distance function based on the symmetrized likelihood between their respective Hidden Markov Models. Dominey and Boucher [5] derive perceptual primitives of contact (touch, push, take, give) as a predicate representation from observed visual scenes. Together with a speech to text transcript of the narrated event they generate a well-formed $< sentence, meaning >$ pair from which they learn their grammatical construction models.

Overall, there is little work so far that attempts to develop models that endow robots with the capability to acquire more complex linguistic constructions that are grounded in representations of action and are acquired in parallel as the linguistic knowledge is acquired.

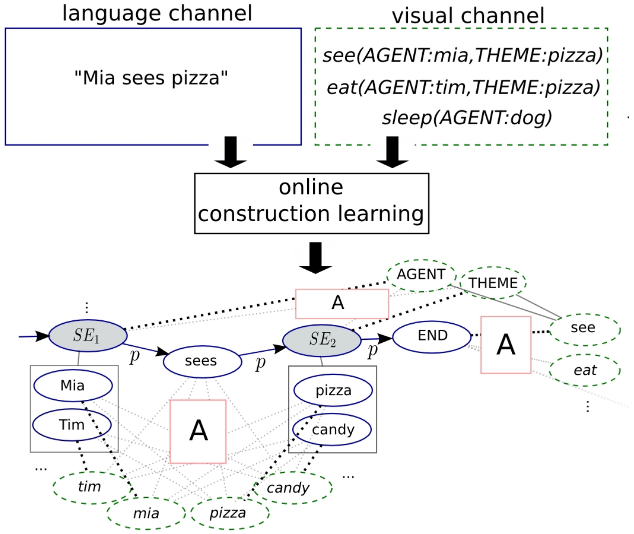
III. Method

In this paper, we attempt to ground linguistic knowledge of a computational model for the induction of syntactic constructions from symbolic input in qualitative action models. In the following, we will first briefly describe the existing model and subsequently the proposed extension.

A. *Learning syntactic constructions*

The existing computational model is represented as an interrelated network and acquires a lexicon and syntactic constructions in an online fashion by observing input examples represented in symbolic form. More specifically, the input comprises two temporally paired channels: a language channel and a visual channel. The language channel presents *NL* utterances to the computational learner, while the visual channel presents a symbolic description of the visual context, which comprises a set of actions taking place while the utterance is being uttered; this reflects natural settings faced by an infant or a robot operating in some environment. Each action $mr_i \in$ *MR* is represented by means of predicate logic formulas, comprising a predicate ξ along with a set of thematic relations. The learning process and an example of a verb-specific construction stored in the network are shown in Fig. 1.

Fig. 1. Overview of the learning process for the existing model. The figure shows the two input channels together with an example construction stored in the network.



The learned network can be roughly divided into a subnetwork representing lexical and a subnetwork representing syntactic constructions, where the syntactic subnetwork builds on the lexical subnetwork and is divided into two sublayers: a slot-and-frame (S&F) pattern layer and a mapping layer. The **lexical subnetwork** encodes simple phrases, i.e. (short sequences of) words, as nodes together with the associated semantic referents, e.g. the word “Tim” and the corresponding semantic referent *tim* in Fig 1. The **S&F pattern layer** represents syntactic constructions as sequences of nodes that together constitute a path. Paths can contain variable nodes that represent slots in the syntactic pattern. These slots can be filled with elements contained in specific groupings. This layer also encodes the associated semantic frames. For instance, in Fig. 1, a syntactic construction is represented as a path p which expresses a pattern "SE_1 sees SE_2", where SE_1 and SE_2 represent syntactic slots in the pattern, which can be filled with groupings of elements such as "Mia" and "Tim" in the case of SE_1 or "pizza" and "candy" in the case of SE_2. The semantic frame associated with the pattern is *see(AGENT,THEME)*. The **mapping layer** contains networks representing construction-specific argument mappings between syntactic patterns and semantic frames together with mappings of the syntactic arguments to semantic arguments. For example, in Fig. 1 an individual mapping network captures the correspondences between SE_1 and *AGENT* as well as SE_2 and *THEME*.

The network contains nodes of two types: Nodes representing **linguistic entities** such as i) simple phrases, e.g. "Tim" or "the cat", ii) syntactic patterns, e.g. "SE_1 sees SE_2", and iii) syntactic slots of constructions represented as sets of elements containing all the simple phrases that can fill the slot, e.g. *SE*$_1$ = [Mia $\rightarrow$ mia, Tim $\rightarrow$ tim]) and nodes representing **semantic entities** such as i) simple semantic referents, e.g. *tim*, ii) semantic frames, e.g. *see(AGENT,THEME)*, and iii) argument slots of frames, e.g. *AGENT*). Correspondences between linguistic nodes and semantic nodes, i.e. form-meaning mappings, are established by capturing their co-occurrence frequencies across observed examples/situations; we apply associative networks [18] to establish associations between form and meaning.

During learning, input examples are processed one-by-one, causing immediate changes in the network structure. Learning is roughly divided into two learning steps: i) update of the lexical layer, where connections between lexical units and semantic referents are established and reinforced, and ii) update of the construction layer, where the model mainly attempts to merge paths, and thus generalizes over specific linguistic and action examples observed. For generalization, the model exploits type variations that have a semantic implication to generalize observed *NL* sentences and (partially generalized) patterns to more abstract patterns. Consider the following example: A learner hears "Mia eats" and "Peter eats" in the above-mentioned visual context. To learn across situations, the model would use its knowledge that the linguistic phrase "Mia" refers to the semantic entity *mia* and that the phrase "Peter" refers to the semantic entity *peter* to bootstrap that the type variation in the sentences' first position ("Mia" vs. "Peter") reflects the meaning difference in the *AGENT* role of *eat*. The model would use its knowledge to acquire the more general pattern shown in (1), where *SE*$_1$ = [Mia $\rightarrow$ mia, Peter $\rightarrow$ $peter$].

(1)

Syntactic pattern	*SE*$_1$ eats
Semantic frame	*eat(AGENT)*
Mapping	*SE*$_1$ $\rightarrow$ *AGENT*

Given an input *NL* sentence, the model finds a meaning by searching the network for corresponding paths/lexical nodes and ranking all possible meanings based on the weights stored in the associative networks. An *NL* sentence is parsed by first replacing units contained in groupings expressing syntactic slots (e.g. Mia) by the set (e.g. *SE*$_1$). Then, the model determines the semantic frame that corresponds to the path in the graph, if such a path exists. Finally, the model retrieves the meanings of lexical units at positions of syntactic slots from the lexical network. It uses the construction's mapping, i.e. the mapping specifying that *SE*$_1$ is the *AGENT*, to insert these meanings into the corresponding argument slots in the semantic frame. For details, please see [1].

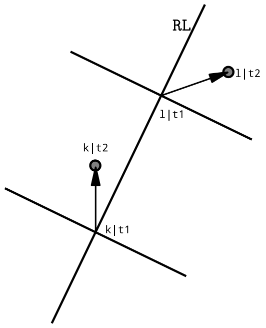


Fig. 2. This figure shows two moving objects k and l at two different time points t_1 and t_2. In this example k is moving towards l at time t_1 on the left hand side of the reference line RL from k to l, l is moving away from k on the left hand side of the reference line from l to k. The corresponding QTC_c relation is $(-+--)$. Reproduced from [19]

B. Learning qualitative action models

In this approach actions are represented as Hidden Markov Models (HMM) over sequences of qualitative relations between the trajector and the landmark. To describe the relative position and movement between the two objects we build on the qualitative trajectory calculus - double cross (QTC_c) [19] as a formal foundation. In general, QTC_c describes the interaction between two movable objects k and l with respect to the reference line RL that connects them at a specific point t in time. QTC_c relations consist of a 4-element state descriptor (C_1, C_2, C_3, C_4) with ternary elements $(+, 0, -)$ yielding a total of $3^4 = 81$ different basic relations. The state descriptor is comprised of the following qualitative relations:

C_1) movement of k with respect to l at time t_1:
- - k is moving towards l
- 0 k is not moving relative to l
- + k is moving away from l

C_2) movement of l with respect to k at time t_1: same as above but with k and l swapped

C_3) movement of k with respect to RL at time t_1
- - k is moving to the left-hand side of RL
- 0 k is moving along RL or not moving at all
- + k is moving to the right-hand side of RL

C_4) movement of l with respect to RL at time t_1: same as above but with k and l swapped

To build the sequence of QTC_c relations we first have to reconstruct the velocities of the two objects from the raw Euclidean positions in the dataset by replaying all movements recorded by the game. As a calculus, QTC_c imposes some limitations on transitions from one state to another, e.g. a transition from $+$ to $-$ has to pass through 0. The resulting sequence misses some of these 0 passages of the state vector elements because the positions were sampled at a fixed rate. The missing intermediate state(s) are added to the sequences one element at a time from left to right. The resulting sequences can contain subsequences where the same symbol is repeated many times. Unlike many spatial reasoning systems, where repeating states are simply omitted, we use a logarithmic compression of repetitive subsequences:

$$|\hat{s}| = min(|s|, 10ln(|s| + 1)) \tag{2}$$

where $|s|$ is the original number of repeated symbols in the sequence and $|\hat{s}|$ is the new number of repeated symbols. By using this compression scheme we preserve information about the acceleration along the trajectory, which increases the overall performance especially for very similar actions like “jumps over” and “jumps upon”. The HMM action models are built using standard K-Means training [20] with iterative Baum-Welch refinement [21]. Merging of two models is done by retraining the HMM on the joined set of observations. The optimal number of hidden states for our models has been empirically determined by searching through the whole parameter space for the maximum likelihood model. Interestingly, it corresponds approximately to the number of distinct QTC_C symbols in the underlying sequences. Classification is done by finding the most probable model to produce the respective sequence.

C. Grounding syntactic constructions in qualitative action models

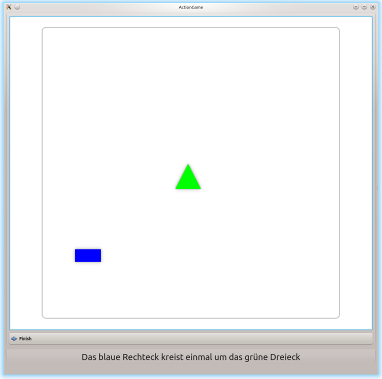
In general, language learning is performed in a similar manner as in the computational model working with completely symbolic meaning representations, albeit sequences/HMMs are considered instead of predicates. For instance, in Example 3, the construction would comprise an HMM instead of the predicate *eat*. However, since in this paper we focus on learning of trajectories/action models, we (still) assume that objects and their roles can be detected out of a visual scene, and these are presented to the learning system in symbolic form. All semantic representations are composed of a trajectory/sequence or an HMM generated from such sequences along with two objects and their roles: *trajector* and *landmark*. For example, an input example could be of the following form:

(3)

NL utterance	red_triangle jumps over blue_circle.
Objects Moves/positions	*trajector*: *red_triangle* *landmark: blue_circle* *move(2389,red_triangle,[10:11]);* *move(2397,red_triangle,[11:12]);. . .*

Notice, however, that observed objects are only given as IDs and that no direct correspondences between words and observed objects or actions are presented. The system must i) establish which words of the utterance refer to the observed objects and which words express the action, and ii) generalize over observed sequences to build action models. Learning problem i) is tackled as in the existing model and described previously, i.e. by searching for variation at the linguistic level which yields corresponding variation at the meaning layer (with the exception that we do not require that predicates for mergeable paths are alike). Learning problem ii) is addressed based on the approach described in Section III-B: For observed input examples, we generate HMM action models. Actions models are merged/generalized in two cases:

1) If they are associated with the same utterance/*NL* pattern, i.e. if they have been observed with the same utterances/patterns.
2) If they are associated with mergeable paths, i.e. if the model determines that utterances are mergeable into more general syntactic patterns, all HMMs associated with mergeable paths are merged as well (and the association scores are accumulated, so that the generalized HMM is directly associated with the resulting generalized path).



Das blaue Rechteck kreist einmal um das grüne Dreieck

Fig. 3. Simple game with two geometric objects which can be freely moved on the gamefield. In this screen test subjects are tasked to revolve the blue rectangle around the green triangle (instruction in the lower part of the screen).

IV. Learning scenario and input data

For evaluation, we consider a learning setting in which the system learns from utterances describing different actions coupled with example trajectories in 2D corresponding to these actions. We considered four actions, i.e. *jump_onto*, *jump_over*, *revolve_around (once)*, and *pushes*. These actions were chosen because they can be performed easily in a 2D-scenario with the same two objects and because they also provide some challenges regarding discriminability, e.g. instances of *jump onto* and *jump over* may have rather similar trajectories.

To collect suitable data, we implemented a simple game in which users could slide geometric objects on a computer screen; an example screen shot is shown in Fig. IV. The game consisted of 100 trials, each corresponding to a single action. In each trial both objects could be moved freely, only for *push* actions collision physics with the landmark was enabled.

In each trial, an utterance expressing an action was displayed on the screen along with two objects named in the utterance, and subjects were asked to perform the action described by the displayed utterance accordingly by sliding the corresponding object(s). Each displayed utterance described one out of the four different actions. For each action a single syntactic pattern was used to generate utterances describing the action, with different combinations of the objects appearing in the syntactic slots of the pattern. For our experiments, we used the following four patterns:

- *trajector* pushes *landmark* from left to right
- *trajector* jumps onto *landmark*
- *trajector* jumps over *landmark*
- *trajector* revolves once around *landmark*

We considered 9 objects for *trajector* and *landmark*, i.e. 3 geometric forms (rectangle, triangle, circle) * 3 colors (red, blue, green), and 25 different utterances (i.e. instantiations of the pattern) were generated for each action, for example "red_circle pushes green_triangle from left to right". For each performed action, we sampled the positions of both objects at a fixed rate. We collected data from 12 subjects (9 male, 3 female, mean age = 29,4 years), yielding 1200 input examples altogether.

V. Experimental evaluation

Since we explore grounded language learning, we are interested in the system's generalization abilities both at the linguistic level and at the visual level. That is, the main goals of the system are to i) understand and generate novel utterances, and to ii) abstract over concrete trajectories of actions, in particular to also recognize actions performed by novel subjects. Thus, we consider two evaluation scenarios:

1) *novel-performer:* 12-fold cross-validation over all subjects, i.e. training on data collected for 11 subjects and testing on the data of the 12-th subject.
2) *novel-utterances:* Data for all subjects are collapsed. We then partition the data into 25 folds by choosing a single (different) utterance for each of the four actions and taking all corresponding utterances as data for one fold. On the partitioned dataset, we perform a 25-fold cross-validation in which all utterances observed during testing are novel, i.e. none of them has been observed during training and thus cannot be understood or generated by performing rote-learning.

For each fold, parameter optimization for the construction learning model, i.e. generalization at the linguistic level, is performed on the training data for that fold prior to testing.[1] We consider three different experiments: two concerning the understanding and one concerning the generation abilities of the developed system. As measures we compute precision, recall and f-measure (the harmonic mean of precision and recall). We compute recall as the percentage of testing examples for which the system generates the correct result and precision as the percentage of correctly generated results of the number of testing examples for which the system actually generates a result (i.e. the system may choose that it cannot determine the result, for instance, because it has not been able to determine a suitable syntactic pattern and/or action model).

In the following, we will first focus on language understanding abilities using a matching and a choosing test, and subsequently explore a language generation experiment. Afterwards, we discuss the achieved results.

A. Matching test

In the first experiment, we evaluate the system's understanding abilities in a matching task. During testing, the system is presented with utterances along with a potential action – i.e. an action which may or may not correspond to the utterance – and is asked whether the action corresponds to the utterance. These testing data are generated such that the action corresponds to the utterance in about 50% of the examples. More specifically, we keep the correct semantics for half of the testing examples and shuffle the semantic representations for the other half such that the semantics do not correspond to the utterance.

Given an utterance, the system makes its choice by first determining a syntactic pattern the utterance is an instantiation of along with the meaning associated with this pattern. If i) the meaning's HMM is the most probable one for the example's sequence/action, and ii) the mapping and objects fit (i.e. the meanings for lexical units appearing in syntactic slots of the utterance appear in the corresponding slots of the example's meaning), the system decides that both match.

B. Choosing test

In this task, we add two distractor meanings to each testing example, thus yielding three potential meanings for an example utterance, out of which exactly one is correct. Importantly, one of them incorporates the same action as expressed by the utterances but different referents, while the other one incorporates the same referents but a trajectory produced for a different action. We then ask the system to choose the action matching the utterance. The system determines whether an utterance and a action match as in case of the previous test, and we count the number of correct choices. This test is not applicable for the *novel-utterances* condition, since in this condition all utterances for each action are alike, and thus no distractor meaning having the same action but different referents can be considered.

C. Language generation test

We evaluate the system's language generation abilities by first extracting a generation grammar based on the learned knowledge induced from the training data and then use this grammar to generate utterances for given testing actions. This grammar is generated by simply extracting all linguistic knowledge – i.e. syntactic patterns, lexical units and groupings of elements – along with their associated meanings and subsequently reversing the associations. For example,

[1] See [1] for details concerning parameters explored in the model for linguistic construction learning. In the experiments presented in this paper we only apply the rating threshold at the word level.

we might extract a pattern "X pushes Y" associated with a meaning comprising an HMM and the information what lexical units can occur in position's X and Y along with their corresponding role in the meaning (such as *trajector*). Given a testing action, the system first determines the most probable HMM for the sequence. Based on the grammar, it can then retrieve the corresponding syntactic pattern along with the information about lexical units and their roles. The generated utterance is considered correct only if it is identical with the example's actual utterance.

For comparison, we create a baseline in which natural language utterances for observed testing meanings are generated by choosing an utterance from the training data which has been observed with a similar meaning representation. In particular, we rate similarity based on both observed referents and action sequences. For this, we evaluate a simple matching strategy using the Levenshtein distance on the compressed QTC_c sequences. We compare the trajectory of the action performance to all trajectories observed in examples of the training data involving the same objects. For all pairs of trajectories t_1, t_2 we calculate a matching score as the normalized Levenshtein distance divided by its theoretical upper bound $lev(t_1, t_2)/max(|t_1|, |t_2|)$. However, this baseline can only yield matches in the *novel-performer* condition; in the *novel-utterances* condition none of the testing utterances has been observed during training and thus cannot be found by simply taking an utterance observed with a similar meaning.

D. Results

Results for all three tests along with their corresponding baseline values are presented in Table I.

TABLE I

RESULTS FOR THE MATCHING (LANGUAGE UNDERSTANDING), THE CHOOSING (LANGUAGE UNDERSTANDING) AND THE LANGUAGE GENERATION TESTS.

Matching test			
Setting	**F$_1$**	**Precision**	**Recall**
Baseline	50% chance		
novel-performer	92.58	92.58	92.58
novel-utterances	87.25	87.25	87.25

Choosing test			
Setting	**F$_1$**	**Precision**	**Recall**
Baseline	∼33% chance		
novel-performer	92.96	99.47	87.25
novel-utterances	*not applicable*		

Language generation test			
Setting	**F$_1$**	**Precision**	**Recall**
Baseline *novel-performer*	89.0	89.0	89.0
novel-performer	87.33	87.33	87.33
Baseline *novel-utterances*	– 0 –		
novel-utterances	77.5	81.66	73.75

The results reveal that the system achieves a large increase in performance over the baseline, i.e. performing at chance, in both language understanding tests. For the matching test, F_1, precision and recall are alike, since the system determines whether the action matches the utterance and if it does not estimate a match votes for *no-match*, thus yielding a result for each testing example if a suitable syntactic pattern and a corresponding action model can be determined. Since most utterances were parsed correctly (as indicated by high values for precision and recall), the system appears to have induced a suitable grammar and action models in most cases, i.e. for most folds. For the *novel-performer* condition, individual values for folds range from 77% to 98%, with only two out of 12 folds yielding values below 90%, i.e. 88% for one fold and 77% for the other. Thus, the learned action models appear to generalize well to a novel performer for most human subjects. For the *novel-utterances* condition values are slightly lower, with values for individual folds ranging from 50% to 97.92%. A value of 50%, i.e. performance at chance, was obtained for a single fold only and likely results from an insufficient determination of syntactic patterns, i.e. syntactic patterns may not have been learned before testing for this fold. However, taken together, the results are promising, showing a large increase in performance over the baseline in both cases. In addition, the system also achieves a large increase over the baseline in the choosing test. In this test, the system also performs with high precision, i.e. performance is close to 100%, indicating that if the system is given several potential meanings for an utterance and cannot determine the correct match it does not confuse the utterance with distractor meanings, even if these are also somewhat similar to the observed utterance, i.e. corresponding to the same action or involving the same objects. Thus, taking the results for both tests together, the learned action models appear to be suitable for determining actions while also yielding a reasonable discrimination ability between different actions.

In the language generation test, the system performs only slightly below the baseline in the *novel-performer* condition, showing that by merging observed action trajectories for several subjects into generalized action models the discriminative power is mostly retained. However, the learned grammar and models yield the additional benefit that the system is able to also generate utterances not observed during training. In particular, in the *novel-utterances* condition the system is still able to generate several utterances correctly, even though it has never observed them or their corresponding meanings before which is an important ability for adaptive artificial agents.

VI. CONCLUSION AND FUTURE WORK

We have presented a system which learns both syntactic patterns and qualitative action models. Learning action models is driven by linguistic observations and syntactic patterns are grounded in these models. We have presented promising results, showing that the proposed system is able to describe novel scenes and to understand novel utterances. In addition, our results indicate that the learned action models generalize well over concrete instances of the same action, while allowing accurate discrimination between different actions. As the learning of syntactic constructions is already tackled in an online fashion, the next step would be to implement the model merging also in an iterative manner. An interesting idea in that directions comes from Stolcke and Omohundro [22], who iteratively train Hidden Markov Models using Bayesian model merging. In the current system, generalization of actions is driven by linguistic information. An interesting point would be to integrate a criterion for merging learned action models based on their similarity and to guide linguistic generalization accordingly. This could, for instance, be useful for detecting synonyms. Further, with respect to application on a robot, one of our main goals is to extend the system to also work with 3D data.

REFERENCES

[1] J. Gaspers and P. Cimiano, "A computational model for the item-based induction of construction networks," *Cognitive Science*, vol. 38, no. 3, pp. 439–488, 2014.

[2] N. C. Chang and T. V. Maia, "Learning grammatical constructions," in *Proceedings of the Annual Conference of the Cognitive Science Society*. Boston, Massachusetts, USA: Cognitive Science Society, 2001, pp. 176–181.

[3] T. Kwiatkowski, S. Goldwater, L. Zettlemoyer, and M. Steedman, "A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings," in *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 234–244.

[4] X. Hinaut, M. Petit, G. Pointeau, and P. F. Dominey, "Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks," *Frontiers in Neurorobotics*, vol. 8, no. 16, pp. 1–17, 2014.

[5] P. F. Dominey and J.-D. Boucher, "Learning to talk about events from narrated video in a construction grammar framework," *Artificial Intelligence*, vol. 167, no. 1-2, pp. 31–61, 2005.

[6] S. Heinrich, C. Weber, and S. Wermter, "Embodied language understanding with a multiple timescale recurrent neural network," in *Proceedings of the 23rd International Conference on Artificial Neural Network*, 2013.

[7] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura, "Learning, generation and recognition of motions by reference-point-dependent probabilistic models." *Advanced Robotics*, vol. 25, no. 6-7, pp. 825–848, 2011.

[8] A. Fazly, A. Alishahi, and S. Stevenson, "A probabilistic computational model of cross-situational word learning," *Cognitive Science*, vol. 34, no. 6, pp. 1017–1063, 2010.

[9] J. M. Siskind, "A computational study of cross-situational techniques for learning word-to-meaning mappings," *Cognition*, vol. 61, no. 1-2, pp. 1–38, Oct-Nov 1996.

[10] D. Roy and A. Pentland, "Learning words from sights and sounds: a computational model," *Cognitive Science*, vol. 26, no. 1, pp. 113–146, 2002.

[11] K. Hsiao, S. Tellex, S. Vosoughi, R. Kubat, and D. Roy, "Object schemas for grounding language in a responsive robot," *Connect. Sci.*, vol. 20, no. 4, pp. 253–276, 2008.

[12] J. A. Feldman, *From Molecule to Metaphor: A Neural Theory of Language*. MIT Press, 2006.

[13] N. Chang, J. Feldman, and S. Narayanan, "Structured connectionist models of language, cognition and action," in *Ninth Neural Computation and Psychology Workshop*, 2004.

[14] R. Van Trijp, L. Steels, K. Beuls, and P. Wellens, "Fluid construction grammar: The new kid on the block," in *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, ser. EACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 63–68.

[15] E. Kruse, R. Gutsche, and F. M. Wahl, "Acquisition of statistical motion patterns in dynamic environments and their application to mobile robot motion planning," in *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 2. IEEE, 1997, pp. 712–717.

[16] A. Droniou, S. Ivaldi, and O. Sigaud, "Learning a repertoire of actions with deep neural networks," in *Joint International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EpiRob)*, 2014, pp. 6–p.

[17] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Modeling manipulation interactions by hidden markov models," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 2. IEEE, 2002, pp. 1096–1101.

[18] R. Rojas, *Theorie der neuronalen Netze*. Springer-Verlag, 1993.

[19] N. Weghe, B. Kuijpers, P. Bogaert, and P. Maeyer, "A Qualitative Trajectory Calculus and the Composition of Its Relations," *GeoSpatial Semantics SE - 5*, vol. 3799, no. Dc, pp. 60–76, 2005.

[20] B.-H. Juang and L. R. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, p. 1639–1641, 1990.

[21] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 02 1970. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177697196

[22] A. Stolcke and S. M. Omohundro, "Hidden markov model induction by bayesian model merging," in *Advances in Neural Information Processing Systems 5, [NIPS Conference]*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 11–18.