A usage-based model for the online induction of constructions from phoneme sequences

Judith Gaspers
Semantic Computing Group
Cognitive Interaction Technology Excellence Center
University of Bielefeld
Email: jgaspers@cit-ec.uni-bielefeld.de

Philipp Cimiano
Semantic Computing Group
Cognitive Interaction Technology Excellence Center
University of Bielefeld
Email: cimiano@cit-ec.uni-bielefeld.de

***Abstract*—According to usage-based approaches to language acquisition, linguistic knowledge is represented in the form of constructions as pairings of meaning and form at multiple levels of abstraction and complexity. The emergence of syntactic knowledge in infants is assumed to be a result of the gradual abstraction of lexically specific and item-based knowledge. In this paper, we present a computational usage-based model accounting for the gradual emergence of a network consisting of constructions at varying degrees of complexity given ambiguous input examples of phoneme sequences coupled with a symbolic representation of the visual context. We provide empirical results on the RoboCup dataset, showing that the model can acquire a compact construction grammar which generalizes successfully to unseen data in an online fashion, with one pass over the data.**

I. Introduction

Children acquire language over a relatively short period of time, mastering the essential syntax of their language by the time they enter school. Thereby, it remains rather unclear which mechanisms facilitate generalization over seen input to yield productive patterns that can be used to process and generate sentences never heard before. There is, however, evidence that language learning proceeds *incrementally*, from simpler to more complex structures [1], as well as – according to usage-based approaches to language acquisition – in an *item-based fashion* [2], [3]. In particular, it is assumed that early on children – unlike adults – maintain an inventory of lexically-specific and item-based constructions which are gradually generalized by replacing concrete lexical items by slots which can be filled by (a restricted group of) words or short sequences of words [2]. Furthermore, usage-based approaches assume linguistic knowledge to be represented in terms of form-meaning pairings – constructions – at varying degree of complexity and generality, e.g. morphemes, words as well as fully productive linguistic patterns. These constructions are captured by an interrelated network – a so-called *construct-i-con* – which comprises both item-specific information and generalizations [4]. In this paper, we explore how the gradual emergence of an inventory containing verb-specific linguistic patterns by an item-based induction of slots can be modeled computationally. Specifically, we introduce a model which captures linguistic knowledge by an interrelated network of constructions at varying degree of complexity. Thereby, we assume that at the modeled stage of learning the child is able to extract sequences of phonemes from the speech signal as well as information in some structured form from the visual context. Like a child, the model learns by observing natural utterances – sequences of phonemes – in a noisy and ambiguous context in which several actions take place, and which in our model is represented through predicate logic formulas. As we model a stage in learning where linguistic patterns emerge gradually, we consider two types of constructions: (short sequences of) words and *'slot-and-frame patterns'* [5]. The model learns these in an incremental fashion in the sense that it first learns the structures of low complexity (words), and then uses these to learn more complex linguistic patterns. This seems also cognitively plausible as children first learn the meaning of (proper) nouns and afterwards of more complex syntactic constructions [1]. Importantly, our model proposes uniform mechanisms for the induction of the different types of constructions. In the language learning process, the model starts with an empty network. While learning proceeds, the network is continuously augmented and refined, dynamically adapting the model to new input. An important aspect of our model is the fact that learning proceeds online, i.e. each example directly causes an update of the network. We provide empirical results on the RoboCup dataset [6], showing that our model can acquire a domain-specific construction grammar with one pass over the data.

II. Related Work

Several computational models have been proposed for the task of word segmentation and/or acquisition of word meaning as well as for the induction of syntactic constructions. In contrast, we explore all three tasks in a single network model where learning at all three levels is interleaved and proceeds gradually and online. With respect to word segmentation, research has mainly focused on utilizing statistics concerning syllable and phoneme regularities [7], e.g. by applying Bayesian methods. Furthermore, several models inferring word-to-meaning mappings have been proposed (e.g., [8], [9]). For instance, Fazly et al. [8] introduced a probabilistic model building on the idea of cross-situational learning and Horst et al. [9] explored a Hebbian Normalized Recurrence Network. Additionally, approaches have been proposed which address both segmenting a speech stream and establishing word-to-

meaning mappings (e.g., [10]). In contrast to approaches addressing word segmentation and/or word-to-meaning mapping, we focus on the extension to complete syntactic constructions and on the interplay between their acquisition and the acquisition of words. Different models have been proposed concerning the acquisition of constructions (e.g [11], [12]). For example, Alishahi and Stevenson [11] introduced a Bayesian model for the acquisition of abstract verb argument structure constructions, assuming the acquisition of words and verb-specific constructions as already solved. In contrast, Chang et al. [12] presented an approach based on Bayesian model merging, where – as in our model – more complex grammatical structures are induced based on previously acquired lexical mappings. In previous work [13], we proposed an algorithm for the induction of constructions which yielded a construction grammar by applying incremental learning steps. Thereby, learning steps were executed as consecutive steps of batch learning, yielding complete constructions only as the result of the last learning step. In contrast, in this study we focus on the interplay between constructions at different levels of complexity in a network and their concurrent acquisition where learning steps are interleaved, yielding an online algorithm. However, there are some commonalities as well between the approach presented here and the previous approach [13]. In particular, in both systems the induction of more complex constructions requires the acquisition of more simple constructions. Further, both incorporate the idea of inducing equivalence classes by searching for sets of substitutable elements and subsequently inspecting if they can account for a slot in a predicate. Kwiatkowski et al. [14] proposed a model for language acquisition which – like our model – works on ambiguous input. The model acquires language in an online fashion by training a non-parametric Bayesian model. However, in contrast to these approaches, our model is represented as a single network and uses phonemic transcriptions as input rather than words. Our work is also similar to the field of semantic parsing. While work in this area has mainly focused on building systems which are trained on examples constituting NLs along with their (manually annotated) corresponding meanings, semantic parsers exist which can handle ambiguous training data. For example, Chen et al. [15] extended several semantic parsing systems to handle ambiguous training data, and Börschinger et al. [16] accomplished the task by inducing a Probabilistic Context Free Grammar. However, those parsers work by iterating over the full training data several times in batch mode which is cognitively implausible and computationally expensive.

III. Model

A. Input and goal

Our model learns analogously to a child by observing natural utterances (NL, represented as sequences of phonemes) in a noisy and ambiguous context (MR, represented by formulas in predicate logic mr). In particular, the input to our model consists of a list of pairs comprising NL utterances coupled with a set of meaning representations, i.e. $(NL, MR = \{mr_1, ..., mr_n\})$. For each example, NL consists of a sequence of phonemes. Each mr_i consists of a predicate ξ and a list of arguments $arg_1, \ldots, arg_n$ (which might be empty). We distinguish between an observed mr and its corresponding template mr^g which is derived from mr by replacing the values of its argument slots by $ARG_1, \ldots, ARG_k$, where k is the number of arguments in mr. We also say that mr instantiates mr^g. Input of the desired form is for instance provided by the RoboCup Soccer corpus [6], which consists of the annotated RoboCup finals from 2001-2004. In this corpus, game events are represented by mrs. The games were commented by humans, constituting the NL utterances. Each NL comment is coupled with a set of meaning representations MR, where NL corresponds to at most one $mr_i \in MR$. To model learning from phoneme sequences, we used a speech synthesis system (i.e. MaryTTS [17]) to transcribe the NLs phonemically. Subsequently, we removed all spaces and markers of word boundaries, yielding unsegmented phoneme sequences. Given a set of examples $\{e = (NL, \{mr_1, ..., mr_n\})\}$, our goal is the induction of a construction grammar, i.e. a set of form-meaning pairings $\{(\hat{NL}, \hat{mr}, \Phi)\}$, represented in terms of a network where linguistic knowledge evolves over the course of time. In particular, we attempt to segment the streams of phonemes into meaningful sequences, i.e. phoneme sequences representing (sequences of) words(s) which map to semantic referents. In the following, such sequences are also referred to as (potential) lexical units. Based on this information, we attempt to induce syntactic patterns such as "X passesto Y". Thus, two types of constructions are learned and represented in our network: (1) constructions at the word level L_W where the form $\hat{NL}$ corresponds to a lexical unit (note that lexical units are not given a priori but must be segmented from the continuous stream of phonemes) and the meaning $\hat{mr}$ to a single semantic referent, and (2) constructions at the complete construction level L_C where $\hat{NL}$ corresponds to a NL pattern and $\hat{mr}$ is represented by exactly one template mr^g. If $\hat{NL}$ contains equivalence classes, these are associated to argument slots in $\hat{mr}$ by a one-to-one mapping Φ: $ECs(\hat{NL}) \rightarrow args(\hat{mr})$. Taking for instance the input examples ("purpletenkicks", $\{ballstopped, badPass(pink1, purple10), pass(purple10, purple7), playmode(play\_on), kick(purple10)\}$) and ("pinkgoalie-kicks", $\{pass(pink1, pink5), kick(purple1)\}$) (note that for the sake of simplicity, in this paper we represent NLs as sequences of characters instead of phonemes), at L_W we would like to induce the form-meaning pairings

(1)

$\hat{NL}$	purpleten
$\hat{mr}$	purple10

$\hat{NL}$	pinkgoalie
$\hat{mr}$	pink1

and at L_C we would like to induce the form-meaning pairing

(2)

$\hat{NL}$	EC1 kicks
$\hat{mr}$	kick(ARG1)
Φ	EC1 $\rightarrow$ ARG1

where the equivalence class $EC1$ = [purpleten $\rightarrow$ $purple10$, pinkgoalie $\rightarrow$ $pink1$] groups the L_W constructions.

B. Representation

Our network model incorporates two basic components: 1) associative networks, and 2) a directed graph.

1. Associative networks: We use associative networks as suggested by Rojas [18] to establish correspondences between form and meaning where connections between neurons which are active concurrently (i.e. between neurons representing form and meaning being observed concurrently) are strengthened, capturing the co-occurrence of frequency between form and meaning. In particular, an associative network A consists of two layers of neurons x and y fully connected by a matrix W of learnable weights. Associations are retrieved from the network by $y = Wx$ and $x = W^T y$. To train the weights, we use the adjusted learning rule suggested by Schatten [19]:

$$\Delta w_{ij} = \eta(x_i - x'_i)(y_j - y'_j) \tag{3}$$

where x'_i and y'_j denote the network's current value of x_i and y_j after processing the input y and x, respectively, and η denotes the learning rate. We refer to the *update* of all weights in A by $w_{ij} = w_{ij} + \Delta w_{ij}$ as $A.update(a_x, a_y)$, where a_x and a_y denote the sets of neurons currently being active in x and y, respectively. Their activation is set to 1, while the activation of all other neurons is set to 0. Further, we say that a $y_j \in y$ is *associated* to a $x_i \in x$ if it maximizes the value of the weights between x_i and all $y_j \in y$.

2. Directed graph: We use a directed graph to capture the segment order of NLs in a similar way as the ADIOS algorithm [20]. Specifically, we represent the NLs of constructions at L_C as indexed paths, where each node corresponds either to a sequence of phonemes, an equivalence class, or marks the start or end of a sequence. Note however that we propose a different strategy than ADIOS for the induction of (generalized) patterns. Specifically, in contrast to ADIOS, which induces linguistic patterns from raw text, we additionally utilize information derived from the visual context, and only merge NLs if a coherent meaning can be established for the resulting generalized pattern. Details on the generalization procedure will be provided in section III-D. Merging a set of mergeable (see section III-D for a definition) paths P of length p_L represented on a directed graph W is referred to as $W.merge(P)$, and yields a single path p_{com} representing the merge of paths in P. The combined path p_{com} is computed by iterating over the nodes for all paths in P concurrently. If all paths are alike at a position pos, the node at position pos in p_{com} is set to that node. Otherwise, it is set to a new node n_{se} representing an equivalence class. Furthermore, for each path in P, the node at position pos is added to the equivalence class and subsequently replaced by n_{se} for each path in W. Finally, all paths in P are deleted from the graph. During the merging procedure new equivalence classes are induced. If a newly induced equivalence class has an element (v_{nl} and/or a v_{mr}) in common with at least one of the already existing equivalence classes, the corresponding equivalence classes are merged into a single equivalence class ec. The nodes corresponding to subsumed equivalence classes are then replaced by the node corresponding to ec. In the model, associations between paths and mr templates are modeled by an associative network A_C. The weights for p_{com} are initialized by summing up the weights contained in A_C of the rows for the subsumed paths[1]. If p_{com} contains equivalence classes, an associative network representing the mapping A_{p_{com},mr^g} is included for each mr^g in L_C which contains slots. Each A_{p_{com},mr^g} is then initialized by adding up weights contained in subsumed associative networks.

The proposed network architecture is illustrated in Fig. 1. As

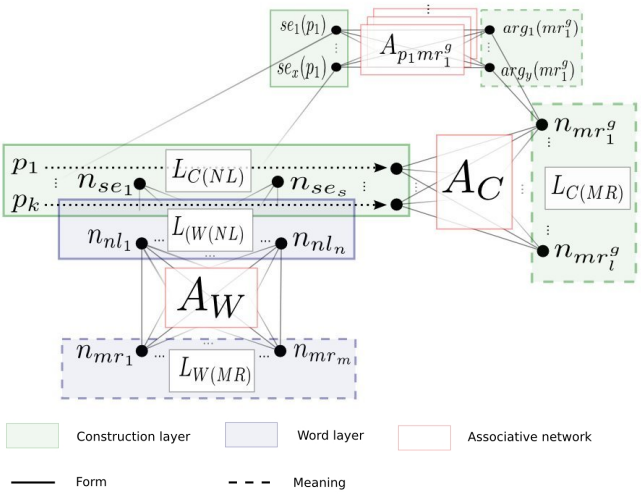

Fig. 1. Network modeling three levels of association

our aim is to include both constructions at the word level L_W as well as constructions at the complete construction level L_C into our network, it is divided into two subnets representing constructions at L_W and constructions at L_C, where L_C builds on L_W. Both subnets consist of a layer representing the form ($L_{W(NL)}, L_{C(NL)}$) and a layer representing the meaning ($L_{W(MR)}, L_{C(MR)}$). Correspondences between form and meaning are modeled by associative networks A_W and A_C. During training, all observed linguistic input is incorporated into the form layers, while the action input is incorporated into the meaning layers. In L_W, each observed lexical unit is modeled as a single node n_{nl}, and semantic referents are modeled as single nodes n_{mr}. Constructions in L_C are modeled as paths through a directed graph ($L_{C(NL)}$). The directed graph incorporates nodes from $L_{W(NL)}$, nodes representing the start n_{START} and the end n_{END} of a sequence, as well as a node n_{se} for each induced equivalence class (these nodes group in turn sets of nodes from $L_{W(NL)}$). $L_{C(MR)}$ contains a node n_{mr^g} for each template derived from mrs observed in the input. In L_C, constructions may include a mapping which maps equivalence classes to argument slots for a specific path p and template mr^g. These mappings are each modeled by an associative network A_{p,mr^g}. As our dataset contains several NL expressions which have no semantic correspondence according

[1]In our current implementation, weights are restricted to values between 0 and 1. Greater values are set to 1, smaller values to 0.

to the underlying PL representation, we include a special node $n_\perp$ in each associative network that allows to capture the fact that a certain linguistic construction has no correspondence at the meaning layer. An example for a concrete L_C construction

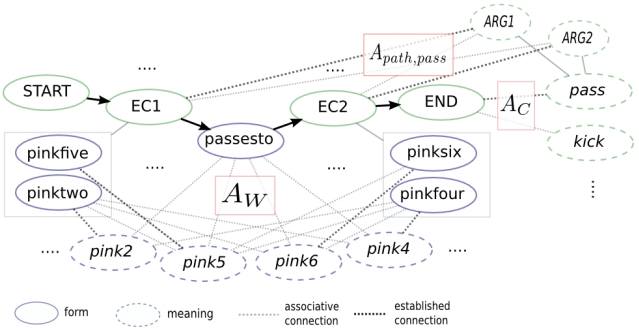

Fig. 2. Example of a construction in the network

is depicted in Fig. 2. It consists of the generalized path $p = (START, EC1, passesto, EC2, END)$ associated via A_C to the template $pass(ARG1, ARG2)$. Its equivalence class nodes EC1 and EC2 are associated via $A_{path,pass}$ to the slots $ARG1$ and $ARG2$, respectively, modeling the construction $\hat{NL}$: "EC1 passesto EC2", $\hat{mr}$: $pass(ARG1, ARG2)$, Φ: EC1 $\rightarrow ARG1$, EC2 $\rightarrow ARG2$.

C. Rating and retrieval of constructions

Given $nl \in x$ (lexical unit or NL) without equivalence classes, the rating for each $mr \in y$ is computed as

$$rating(nl, mr) = A.w_{nl,mr}. \tag{4}$$

In case of a pattern $nl \in L_{C(NL)}$ containing equivalence classes the rating for each $mr \in y$ is computed by augmenting the weight $A_c.w_{nl,mr}$ with the association scores between each $se \in ses(nl)$ and its associated slot $associated(se)$ in mr as

$$rating(nl, mr) = A_C.w_{nl,mr} + \sum_{se \in A_{nl,mr}.x} A_{nl,mr}.w_{se,associated(se)} \tag{5}$$

if a one-to-one mapping between the equivalence classes $ECs(nl)$ and the argument slots $args(mr)$ exists. Otherwise, the *rating* is set to 0. If a single $mr' \in y$ maximizes $rating(nl, mr')$ as defined by equations 4 or 5 for a given form $nl \in x$ it is said to be the *meaning* of nl. If it is additionally the case that $rating(nl, mr') > \theta_R$, then nl is regarded as *learned*.

Given $nl \in L_{W(NL)}$, we can determine whether a (learned) meaning exists and if so retrieve the meaning as described above. In order to retrieve the meaning of a complete NL, we first replace subsequences contained in an equivalence class by the equivalence class, and subsequently NL is segmented at boundaries of equivalence classes. If the resulting sequence of segments is contained in $L_{C(NL)}$, we can again determine whether a meaning exists and if so retrieve it. If no meaning with a $rating$ score greater then zero exists, the meaning is set to $\perp$. If the meaning is $\perp$ or if no corresponding pattern can be found, NL cannot be understood (parsed) by the model. Otherwise the meaning of each sequence at a position corresponding to an equivalence class is retrieved from L_W and inserted into the slot associated to the equivalence class (via the associative network forming the corresponding mapping) in the retrieved template.

D. Generalization

Generalization is performed in essence by i) inducing equivalence classes and ii) merging paths to more general and productive ones. As equivalence classes group elements whose exchange in a NL pattern causes a change in the corresponding meaning with respect to an argument slot, they are identified by searching for differences in patterns which also account for a distinction in the corresponding meaning. In particular, we explore an idea from previous work [13] stating that sets of substitutable elements in a group of NLs represent an equivalence class if they can account for the difference in the argument of a given slot. Given for instance the form-meaning pairings depicted on the left side of the arrow in the following example, one can easily infer the correspondences shown on the right side.

(6)

pinktwo kicks	pinkone kicks	→	X kicks
$kick(pink2)$	$kick(pink1)$		$kick(ARG1)$
			X $\rightarrow ARG1$

The inference that "pinktwo" and "pinkone" are substitutable and that the grouping accounts for the slot in the corresponding predicate $kick$ can be performed based on two observations: 1) the NLs differ in one position/slot pos and the corresponding mrs differ in one argument position/slot ARG, and 2) the meanings of the observed segment at position pos occur in argument slot ARG for both examples. Note that the second condition is crucial as our goal is to develop a model which handles noisy input. However, in our model we need to compare an example $e = (NL, \{mr_1, ..., mr_n\})$ with a path p in the network and therefore the previous observations cannot be implemented directly. Instead, we adapted the described conditions as follows. Given an example e and a path p we retrieve an mr^g associated to p from A_C as the corresponding meaning; p is then *mergeable* with a segmented NL s if s and p differ in at most k positions and 1) the element at each differing position pos in s corresponds to a lexical unit whose learned meaning is also observed in a slot of one observed $mr_i \in \{mr_1, ..., mr_n\}$ and mr_i instantiates mr^g, and 2) the element at each such position pos in p corresponds to either a lexical unit holding a learned meaning or an equivalence class.

E. Language learning algorithm

Language learning starts with an empty network and then proceeds online, i.e. each observed example $e = (NL, \{mr_1, ..., mr_n\})$ directly causes an update of the network. The algorithm is roughly divided into two learning steps: 1) acquisition of words, and 2) acquisition of constructions.

1. Acquisition of words: While infants may use several cues in order to segment words out of the speech stream, in this work we explore how meaningful sequences of phonemes can be segmented out of a continuous stream based solely on

context information, and thus word segmentation and word to meaning mapping are to a great extent interleaved. In particular, given an example $e = (NL, \{mr_1, ..., mr_n\})$ the algorithm starts by extracting all lexical units, i.e. sequences potentially mapping to a semantic referent, from NL. In this work, we simply regard all subsequences of length l_{min} to l_{max} as lexical units. Particularly, we are interested in subsequences mapping to arguments as in the subsequent generalization steps these sequences might be grouped into equivalence classes, thus yielding a valuable basis for generalization. Therefore, in addition to the subsequences all arguments $args$ are extracted from e, and the co-occurrence between all extracted sequences and arguments is captured by training the weights in A_W by $A_W.update(units, args)$. During several observations of a sequence together with its referent and execution of corresponding updates in A_W, a (learned) meaning may be established between both.

2. Acquisition of constructions: While in step 1 several subsequences are extracted, in step 2 our first goal is to segment NL at boundaries of sequences mapping to arguments. This is essential in order to induce correct slots in NLs/patterns and thus to avoid subsequent generalization errors. For example, given the two NLs "pinktwokicks" and "pinkfourkicks" and a corresponding incorrect segmentation "pinktw okicks" and "pinkfou rkicks" it is not possible to induce a pattern "X kicks". The basic idea for segmenting an utterance NL is to identify sequences of phonemes mapping to arguments observed concurrently. Thereby, we regard a lexical unit (subsequence of NL) as mapping to an argument if its learned meaning is observed concurrently in an argument slot in e. For any observed argument, several sequences statisfying the criterion may exist. For example, if a sequence "pinkeleven" maps to an argument $pink11$, so may subsequences of "pinkeleven" such as "inkeleven", "nkeleve", etc. We therefore take the length of sequences into account in case of several sequences mapping to the same argument with an equal weight by taking the longest sequences in that case. Altogether, given an example e, we search for the subsequence in its NL fulfilling these criteria for every observed argument in $\{mr_1, ..., mr_n\}$. If sequences are contained in equivalence classes, these are replaced beforehand by the equivalence class. NL is then segmented at the boundaries of the identified sequences, yielding a sequence s of segments. Subsequently, the model searches for paths contained in the network which are mergeable with s, and if mergeable paths $p_{mergeable}$ exist, s is merged with them by $L_{C(NL)}.merge(\{s, p_{mergeable}\})$ into a new path. If no such paths exist, s is incorporated as a new path. Whichever applies, a new path $path$ is incorporated into the network, and all templates mrs observed concurrently in e are extracted from e and if not yet present included into the network as new nodes. Subsequently, the co-occurence between mrs and $path$ is captured by $A_C.update(path, mrs)$. If $path$ contains equivalence classes, an associative network modeling a mapping is incorporated for each template containing slots. Subsequently, our algorithm updates mappings between $path$ and mr templates observed in e by iterating over all nodes in $path$. For each node corresponding to an equivalence class, the phoneme sequence at the corresponding position pos_{se} in the example's segmented NL is inspected. Specifically, for each template mr_e derived from the input sequence, it is determined if the lexical unit's meaning is observed in a slot ARG of mr_e. If so, the correspondence between pos_{se} and ARG is captured by an update of the corresponding associative network by executing $A_{path,mr_e}.update(pos_{se}, ARG)$. Otherwise the fact that the lexical unit's meaning is not observed in an argument slot is captured by $A_{path,mr_e}.update(pos_{se}, \perp)$. Due to the fact that through merging paths may become identical by replacing nodes with newly induced equivalence class names, as a final step our algorithm merges all identical paths.

IV. Evaluation

The main goals in generalizing observed examples are 1) to enable the model to use constructions in a compositional manner, thus allowing understanding/generation of novel phoneme sequences, and 2) to keep the network size – and therefore the corresponding grammar – small. We evaluated the system's abilities concerning both goals on the RoboCup corpus. In particular, we evaluated our model on a semantic parsing task; semantic parsing is the task of mapping NL sentences to $\hat{mr}$s. The corpus contains 4 Robocup games. While the training data is ambiguous, the reference corpus (gold standard) is disambiguated and contains one meaning representation for each utterance. Recently, Chen et al. [15] evaluated several semantic parsers on the corpus by using 4-fold cross validation on the 4 games, where training was done on the ambiguous training data, while the gold standard was used for testing. They computed precision (the percentage of $\hat{mr}$s produced by the system that were correct) and recall (the percentage of $\hat{mr}$s that the system produced correctly), and presented results by means of the F_1 score (the harmonic mean of precision and recall) [15]. We applied the same evaluation scheme, albeit using the phonemically transcribed sequences (k = 2, l_{min} = 6, l_{max} = 15). Our algorithm incorporates a threshold θ_R which was utilized in case of A_W. We optimized this parameter for each fold by training the model with varying parameters on the ambiguous training data and subsequently measuring its performance on the gold standard corresponding to the training data (note that test data was not used during parameter optimization). Both for parameter optimization as well as for testing, each example was only presented once to the system. Without performing generalization a learner may at most understand NLs which were presented during training. We therefore compared our model's F_1 score to the F_1 score that would be achieved if the model would have rote learned the meaning for each observed example to estimate its generalization abilities (note that rote learning is not possible in case of most NLs as the data is ambiguous). While an F_1 score of 40.2% was obtained with a naive rote learning baseline, our model achieved an F_1 score of 81.1% (precision: 95.8, recall: 72.2), thus performing very well on unseen data. Our learning and generalization mechanisms are thus effective

as the learning proceeds online with only one pass over the data, especially given the fact that NLs observed early on cannot be generalized as segmentation is not yet possible due to the unavailability of lexical units holding a learned meaning. Averaged over all folds, 730.25 individual phoneme sequences (types) are contained in the RoboCup data set while our model extracted 395.25 NL patterns averaged over all folds. Generalization thus produces a compact grammar; the number of derived patterns is much smaller than the number of examples observed. Table I indicates the number of average patterns derived for each predicate in the RoboCub dataset and gives examples of particular patterns which have been induced. As indicated by the high precision of 95.8%, the established associations between NL patterns and predicates and equivalence classes and argument slots were mostly correct. Averaged over all folds, 182.5 patterns were extracted which

TABLE I
EXAMPLES AND NUMBER OF PATTERNS AVERAGED OVER ALL FOLDS

associated meaning	avg #patterns	Example of an extracted $\hat{NL}$
$pass(P,P)$	95.25	SE passesto SE nearmidfield
$kick(P)$	42	SE dribblestowardthegoal
$badPass(P,P)$	46	SE loosestheballto SE
$turnover(P,P)$	19	SE turnstheballoverto SE
$steal(P)$	8.75	SE stealstheball
$block(P)$	1.5	SE blockstheball
$playmode(PM)$	0.25	pink SE
$defense(P,P)$	0	–
$ballstopped$	0	–
$\perp$	182.5	pinkteam willkick in

were regarded as meaningless, and in fact about one fifth of the comments in the games actually does not have a correct meaning according to the semantic representation in predicate logic [6]. Yet, the model also judged several patterns incorrectly as meaningless. In particular, $playmode$ events were often associated to $\perp$ since the model is not able to induce correct slot-and-frame patterns in case of $playmode$ due to the fact that instances of its argument are composed of several individual parts which in turn correspond to both the predicate and an argument; an example describing a $playmode$ event taken from the gold standard is given by ("freekickfromtheopurpleteam", $playmode(free\_kick\_l)$). As can be seen, the whole NL maps to the complex argument and therefore a correct slot-and-frame pattern cannot be derived. Furthermore, patterns containing more equivalence classes than required by the appropriate predicate – e.g. "SE1 triestopassto SE2 butwasinterceptedby SE3" and its corresponding predicate $badPass$ – cannot be learned by our model due to the fact that no one-to-one mapping can be extracted in this case.

V. CONCLUSION

We have presented a model for the acquisition of constructions at different levels of complexity, from (sequences of) words through to generalized patterns given ambiguous input examples of phoneme sequences coupled with a symbolic representation of the visual context. Linguistic knowledge is acquired in an incremental, usage-based and online fashion. We have tested the model on the Robocub dataset, showing that it can indeed acquire a domain-specific construction grammar effectively, with one pass over the data. This is contrast to many other approaches which perform learning in batch mode and require several passes over the data. To our knowledge, our model is the first that induces syntactic patterns by starting from phoneme sequences. A particular feature of our model is that it learns structures of different complexity as well as the syntax and semantics in an interleaved and parallel fashion.

ACKNOWLEDGMENTS

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

REFERENCES

[1] P. Bloom, *How Children Learn the Meanings of Words*. MIT Press, 2000.
[2] M. Tomasello, N. Akhtar, K. Dodson, and L. Rekau, "Differential productivity in young childrens use of nouns and verbs," *Journal of Child Language*, vol. 24, pp. 373–387, 1997.
[3] M. Tomasello, *First Verbs: A Case Study of Early Grammatical Development*. Cambridge University Press, 1992.
[4] A. Goldberg and L. Suttle, "Construction grammar," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 1(4), pp. 468–477, 2010.
[5] J. M. Pine and E. V. M. Lieven, "Slot and frame patterns and the development of the determiner category," *Applied Psycholinguistics*, vol. 18, pp. 123–138, 1997.
[6] D. L. Chen and R. J. Mooney, "Learning to sportscast: A test of grounded language acquisition," in *Proceedings of the 25th International Conference on Machine Learning*, 2008.
[7] L. Pearl, S. Goldwater, and M. Steyvers, "Online learning mechanisms for bayesian models of word segmentation," *Research on Language and Computer*, 2011.
[8] A. Fazly, A. Alishahi, and S. Stevenson, "A probabilistic computational model of cross-situational word learning," *Cognitive Science*, vol. 34(6), pp. 1017–1063, 2010.
[9] J. S. Horst, B. McMurray, and L. K. Samuelson, "Online processing is essential for leaning: Understanding fast mapping and word learning in a dynamic connectionist architecture," in *Proceedings of the Twenty-Eighth Annual Conference of the Cognitive Science Society*, 2006.
[10] D. Roy and A. Pentland, "Learning words from sights and sounds: A computational model," *Cognitive Science*, vol. 26, pp. 113–146, 2002.
[11] A. Alishahi and S. Stevenson, "A computational model of early argument structure acquisition," *Cognitive Science*, vol. 3, pp. 298–834, 2008.
[12] N. C. Chang and T. V. Maia, "Learning grammatical constructions," in *Proceedings of the 23rd Cognitive Science Society Conference*, 2001.
[13] J. Gaspers, P. Cimiano, S. Griffiths, and B. Wrede, "An unsupervised algorithm for the induction of constructions," in *Proceedings of the joint IEEE International Conference on Development and Learning and on Epigenetic Robotics*, 2011.
[14] 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 *Proc. of the 13th Conf. of the European Chapter of the Assoc. for Comp. Ling.*, 2012.
[15] D. L. Chen, J. Kim, and R. J. Mooney, "Training a multilingual sportscaster: Using perceptual context to learn language," *Journal of Artificial Intelligence Research*, vol. 37, pp. 397–435, 2010.
[16] B. Börschinger, B. K. Jones, and M. Johnson, "Reducing grounded learning tasks to grammatical inference," in *Proceedings of the Int. Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.
[17] M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research, development and teaching," *International Journal of Speech Technology*, vol. 6, pp. 365–377, 2003.
[18] R. Rojas, *Theorie der neuronalen Netze*. Springer-Verlag, 1993.
[19] R. Schatten, "Systemic architecture for audio signal processing," in *Proceedings of the 7th European Conference on Artificial Life*, 2003.
[20] Z. Solan, D. Horn, E. Ruppin, and S. Edelman, "Unsupervised learning of natural languages," *Proceedings of the National Academy of Sciences*, vol. 102(33), pp. 11 629–11 634, 2005.