Which Input Abstraction is Better for a Robot Syntax Acquisition Model? Phonemes, Words or Grammatical Constructions?

Xavier Hinaut
Inria Bordeaux Sud-Ouest, Talence, France.
LaBRI, UMR 5800, CNRS, Bordeaux INP, Université de Bordeaux, France.
Institut des Maladies Neurodégénératives, UMR 5293, CNRS, Université de Bordeaux, France.
xavier.hinaut@inria.fr

***Abstract*—There has been a considerable progress these last years in speech recognition systems [13]. The word recognition error rate went down with the arrival of deep learning methods. However, if one uses cloud-based speech API and integrates it inside a robotic architecture [33], one still encounters considerable cases of wrong sentences recognition. Thus speech recognition can not be considered as solved especially when an utterance is considered in isolation of its context. Particular solutions, that can be adapted to different Human-Robot Interaction applications and contexts, have to be found. In this perspective, the way children learn language and how our brains process utterances may help us improve how robot process language. Getting inspiration from language acquisition theories and how the brain processes sentences we previously developed a neuro-inspired model of sentence processing. In this study, we investigate how this model can process different levels of abstractions as input: sequences of phonemes, sequences of words or grammatical constructions. We see that even if the model was only tested on grammatical constructions before, it has better performances with words and phonemes inputs.**

I. Introduction

A. Robots as models to study language acquisition

Robots are interesting for studying language in many perspectives. Some of the long lasting questions are, for instance how languages evolve or emerge [27], [28], how language or symbols in general could be grounded [14], [26] or how the linguistic or non-linguistic symbols may emerge from grounding [31]. In particular, one may be interested to have a robot able to mix vision and dialog interaction in order to vocally command the robot to grasp some objects in complex environments [1], [15], [32]. However, even if some of these systems provide some transparency on how they work[1], they rarely help to understand how our brain processes languages or how children could acquire one. Developmental architectures [12], [19] are inspired from children development and do not require to have all (vocabulary or syntactic) abilities prefixed since the beginning of the learning period. Some studies have used different cognitively inspired frameworks with robotics, such as Embodied Construction Grammar [8] and construction grammar [12], [25]. Our brains process utterances in a robust fashion in a variety of contexts: we believe that the lack of brain-inspiration in these systems results in a gap of robustness with human performance. In our approach, we try to build an architecture that is able to tackle several of these points and get a step closer to the understanding of brain processes, language developmental strategies and symbol grounding.

B. Our question and hypothesis

Considering a system that learns to parse a sentence given a stream of inputs, one question is **what is the optimal level of abstraction of the inputs**[2]**: phonemes, words or grammatical constructions?**

Here, we only compare purely symbolic input representation and do not consider raw acoustic signals or distributed representation of coding, such as word embedding. In particular we want also to see the **robustness to noise of these different representations**. This is particularly important when such a system is used with real speech signals, and have to deal with the misrecognition of words. We previously started a step in that direction by enabling the model to generalize on sentences with unknown (or unrecognized) words [18].

C. Broca sentence parsing model

The neural parser proposes to model how the human brain processes sentences and is inspired from several studies in neuroscience [5], [16]. A schema of how the global architecture is and how inputs are processed can be seen on Figure 1.

The model is an analogy to a sub part of Broca's area (a region of prefrontal cortex, involved for instance in syntax processing) and the striatum (a subpart of the basal ganglia). Both are generally involved in sequence processing and learning, and in particular in sentence parsing.

Because there are probably as many neuro-inspired models as the number of modellers, we want to make clear our claims about this neuro-plausibility and give some high-level

[1] For instance, when they do not include multiple ad hoc "hacks" to make them work in the desired experimental conditions

[2] Given a particular corpus, because one could assume that this could change on the size and complexity of the corpus.

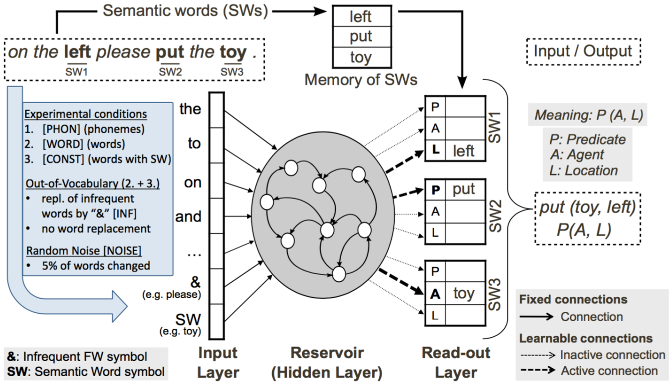
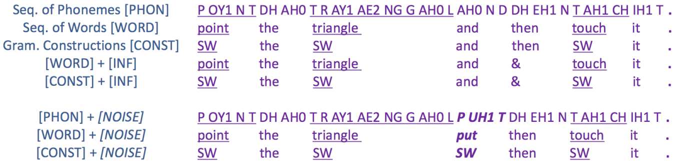


Fig. 1. **Sentence parsing model with different input conditions.** The system processes inputs as follows: (top left) from a sentence as input, the model outputs (middle right) an action command that can be performed by a robot. The processing of the sentence is sequential: each symbol of the sequence (phoneme, word, ...) is given one at a time as a one-hot encoding. The final thematic roles for each SW is read-out at the end of the sentence (but partial predictions can be read-out when the parsing is on-going). Before entering the neural network, the sentence is preprocessed depending on the main condition (PHON, WORD or CONST) and on the optional condition (INF and/or NOISE). Semantic Words (nouns, verbs, ...) are replaced by a SW symbol. Infrequent function words (IWs) are replaced by the *&* symbol. Here, the input layer only represents word symbols, but in the PHON condition these are replaced by phonemes. Example of input sequences for different conditions can be seen in Figure 2. Figure modified from [18].

information about our model in order to easily compare it with other models:

- the computations rely on **distributed units** (leaky average firing rate neurons),
- these **computations are generic and not hand- or task-designed**[3] contrary to some models (e.g. Bayesian)[4]: there is no task a priori applied on the computations,
- it **processes a sentence word by word** instead of taking the whole sentence in one shot or as a bag of words[5],
- it is an **anytime** algorithm: it can give partial results (i.e. predictions) before the end of a sentence,
- it can be **trained in one-shot [16] or incrementally** with a Hebbian learning rule[6],
- the learning algorithm **does not unfold time** contrary to the Backpropagation Through Time (BPTT) algorithm [35], [36],
- we do not use a specific framework for language, on the contrary it is **based on the general Reservoir Computing (RC) paradigm**,
- this RC paradigm is **inspired from neuroscience** and recently inspired neurobiologists in return: e.g. how to decode electrophysiological activity in the prefrontal cortex of a monkey[7].



Seq. of Phonemes [PHON]	P OY1 N T	DH AH0	T R AY1 AE2 NG G AH0 L	AH0 N D	DH EH1 N	T AH1 CH	IH1 T	.
Seq. of Words [WORD]	point	the	triangle	and	then	touch	it	.
Gram. Constructions [CONST]	SW	the	SW	and	then	SW	it	.
[WORD] + [INF]	point	the	triangle	and	&	touch	it	.
[CONST] + [INF]	SW	the	SW	and	&	SW	it	.
[PHON] + *[NOISE]*	P OY1 N T	DH AH0	T R AY1 AE2 NG G AH0 L	***P UH1 T***	DH EH1 N	T AH1 CH	IH1 T	.
[WORD] + *[NOISE]*	point	the	triangle	***put***	then	touch	it	.
[CONST] + *[NOISE]*	SW	the	SW	***SW***	then	SW	it	.

Fig. 2. Symbol sequences given as input to the neural network depending on the conditions. The same sentence (see 2nd line, WORD condition) is given as input in order to see the effect of the different conditions. In PHON cond., a sequence of phonemes is inputed into the network using CMU's dictionary representation: e.g. *point* is replaced by the sequence of symbols "*P, OY1, N, T*". In CONST cond., semantic words are replaced by a SW symbol. In INF cond., infrequent words are replaced by "*&*" symbol: here, *then* is replaced by "*&*". In NOISE cond., 5% of the words are randomly replaced by another one: here, *and* is replaced by *put*.

Several models of sentence acquisition, comprehension or production have been designed [12], in particular models based on neural networks [3], [4], [7], [9]–[11], [23]). However, to our knowledge none of such models combine all these properties, even if not based on RC paradigm. More information about previous versions of the model available in supplementary materials[8].

D. How to deal with Out-of-Vocabulary words?

In a developmental perspective it is important, for a child or a robot, to be able to deal with unknown words. Even if they do not need to infer their meaning yet. In such a developmental approach it is also interesting to not only rely on a fixed vocabulary, but on an evolving one [19]. In speech recognition and natural language processing (NLP) in general, some processed words may be unknown: *i.e.* not in the vocabulary list of the processing system. Theses words are called Out-of-vocabulary (OOV), and are often represented by the marker *UNK* for *unknown*. How to deal with these OOV words is a well-known problem in the field [21]. One can use ad hoc and hand-crafted tricks to deal with such a problem, which seems often the case in HRI, even if not explicitly stated in the papers. On the contrary, we take advantage of the generalization properties of our neural architecture and incorporate the fact that a word was not recognized instead of discarding it (see subsection II-E and [18] for more details).

II. Methods

A. Echo State Networks (ESN)

The neural parser is based on an ESN [20]: a particular kind of recurrent neural network (RNN) in which inputs are projected to a random recurrent layer, and only the output layer (called the "read-out") is modified by learning.

[3] In particular, the computations performed in the input and recurrent layers.

[4] Only input and outputs units are encoded with localist (i.e. one-hot) symbols, but there is no reason to think that a distributed coding should make the model work differently. Because each symbol given as input is already subject to a random projection.

[5] Like a spoken sentence which comes like a flow and is available only once: once has to rely on working memory to keep the information about the whole sentence.

[6] X. Hinaut and S. Wermter, "An incremental approach to language acquisition: Thematic role assignment with echo state networks," in Proc. of ICANN 2014, pp. 33–40.

[7] Neuroscience studies borrowed this idea of decoding non-linear and high-dimensional computations through time.

[8] Supplementary material and source code are available at https://github.com/neuronalX/Hinaut2018_icdl-epirob

The units of the recurrent neural network have a *leak rate* (α) which corresponds to the inverse of a time constant. These equations define the update of the ESN:

$$\mathbf{x}(t+1) = (1-\alpha)\mathbf{x}(t) + \alpha f(\mathbf{W^{in}}\mathbf{u}(t+1) + \mathbf{W}\mathbf{x}(t)) \quad (1)$$

$$\mathbf{y}(t) = \mathbf{W^{out}}\mathbf{x}(t) \quad (2)$$

where $\mathbf{u}(t)$, $\mathbf{x}(t)$ and $\mathbf{y}(t)$ are the input, the reservoir (i.e. hidden) state and the read-out (i.e. output) states respectively at time t. α is the *leak rate*. $\mathbf{W^{in}}$, $\mathbf{W}$ and $\mathbf{W^{out}}$ are the input, the reservoir, and the read-out matrices respectively. f is the *tanh* activation function. After the collection of all reservoir states, the following equation defines how the read-out (i.e. output) weights are trained. In order to prevent from overfitting[9], we use ridge regression (also known as regression with Tikhonov regularization), which probably provides the most stable solution in this context [22]:

$$\mathbf{W^{out}} = \mathbf{Y^d}\mathbf{X^T}(\mathbf{X}\mathbf{X^T} + \beta\mathbf{I})^{-1} \quad (3)$$

where $\mathbf{Y^d}$ is the concatenation of the desired outputs, $\mathbf{X}$ is the concatenation of the reservoir states (over all time steps for all trained sentences), β is the regularization coefficient (called ridge in the remaining of the paper) and $\mathbf{M^T}$ is the transpose of matrix $\mathbf{M}$.

B. Corpus

The corpus was obtained by asking naive users (agnostic about how the system works) to watch several actions in a video and give the commands corresponding to the motor actions performed, as if they wanted a robot to perform the same action. The video used is available online with the first experiments we did with robots [17]. Five users were recruited and each user provided 38 commands: this gives a total of 190 sentences. Note that some sentences provided by users are complex and some times probably ungrammatical (see Table I).

C. Experiments

Based on a previously developed neural parser model [16], [18] we enhanced it in order to make it able to process a given sentence at three different levels of abstraction:

- sequence of phonemes (PHON)
- sequences of words (WORD)
- grammatical construction (CONST)

From these three kinds of inputs we define the three different conditions of our experiments. All inputs are given symbol by symbol, in a one-hot (localist) encoding. Please refer to the supplementary materials and to [16], [18] in order to have more details on the model.

TABLE I
SOME SENTENCE EXAMPLES FROM THE NOISY ENGLISH CORPUS. DIFFERENT TYPE OF SENTENCES ARE GIVEN: 1. SEQUENCE OF ACTIONS 2. IMPLICIT REFERENCE TO VERB 3. IMPLICIT REFERENCE TO VERB AND OBJECT 4. CROSSED REFERENCE 5. REPEATED ACTION 6. UNLIKELY ACTION 7. PARTICULAR FW

TYPE	SENTENCE EXAMPLE
1	touch the circle **after** having pushed the cross to the left
1	put the cross on the left side and **after** grasp the circle
2	**move** the circle to the left **then the cross to the middle**
3	**put** first the triangle on the middle **and after on the left**
4	***push*** **the triangle** and *the circle* ***on the middle***
5	hit **twice** the blue circle
5	grasp the circle **two times**
6	put the cross to the right and **do a u-turn**
7	put **both** the circle and the cross to the right

D. Details on the Phoneme Extension of the Model

In the case of PHON condition, we extended the model [16] to process directly sequence of phonemes instead of sequence of words. Each word is replaced by its corresponding list of phonemes based on the Carnegie Mellon University word-phoneme correspondence dictionary (CMUdict v0.07)[10]. No additional space or any other clue enabling the model to detect boundary of words was added. Each sentence is then encoded as a succession of input unit activations in a localist (i.e. one-hot) encoding: 1 for the corresponding phoneme, and 0 for others. The outputs are generated as in the previous model, as if the grammatical constructions were processed. Thus, the model has no simple cue indicating that a semantic word is being processed (i.e. activation of the SW input unit, as it is the case in the previous model).

E. Infrequent symbols category

For the sequence of words (WORD) and grammatical construction (CONST) inputs we also add a condition where we replace infrequent words by an "&" marker in order to see if this enables better generalization. Before training, we simply replace the most infrequent words in the training corpus. We count the number of occurrences of the words in the training corpus and define a threshold θ ($\theta = 5$). The words that have a lower number of occurrences are replaced by an Infrequent Word (IW) marker "&". This enables our system to smartly process unknown words during test phase [18]. The idea behind is the following: the reservoir is trained to have such IW markers at different positions inside sentences, thus enabling it to not "freak out" when an unknown word appears. This word replacement was not used in the PHON condition for several reasons:

- if we use phonemes we may not have access to word recognition, so the problem of misrecognizing a word does not exists in this case.
- although it could help, the aim is to assess if relying on PHON without speech recognition is possible.

[9] For the current data set, the optimal number of neurons is approximately of 160, 280 and 300 neurons for the WORD, CONST and PHON conditions. Thus we need to regularize if we want to use more units in the reservoir (i.e. 500).

[10] Available at http://www.speech.cs.cmu.edu/cgi-bin/cmudict

F. Dealing with noisy inputs

As we are interested in the performance in real world scenarios of this neural parser, we would like the system to be able to handle and recover from speech recognition errors. In Human-Robot Interaction studies, it is common that this issue is not considered (badly recognized sentences are just discarded) or *ad hoc* methods are applied (*e.g.* researchers write by hand a correspondence dictionary in order replace words that are phonetically close to the desired ones). Even using mainstream APIs, such as the Google Speech API, doesn't prevent this issue (i.e. several words are not correctly recognized) from occurring in many sentences. Such APIs seem to be optimized for web searches, etc., but not for HRI vocabulary. Twiefel et al. [33], [34] have been able to enhance the results from Google API by post-processing its results: by decomposing the sentences obtained in sequences of phonemes and then finding the most probable sentence based on Levenshtein distance. Although the performances of the robotic architecture tested in these studies provide robust results, even in the presence of significant ambient noise[11], we want to enable the core part of the architecture (i.e. the recurrent network) to be intrinsically more robust.

G. Training and testing

For all but noisy experiment conditions, we did a 10-fold cross-validation. For all experiments we averaged over 100 instances if not stated otherwise. In order to evaluate the performance, we record the activity of the read-out layer at the last time step, which is when the final dot is given as input. We first discard the activations that are below a threshold of 0.5. Finally, if there are several possible roles for a particular SW, we do a winner-take-all and keep the role unit with the highest activation.

More implementation details on the training and testing procedures, about simulating noisy speech, and about hyper-parameters are available in the supplementary materials

III. Results

A. Phonemes, words or grammatical constructions?

In Table II are presented the overall results for all the condition combinations. For PHON, WORD and CONST conditions, we obtain generalization errors of 18.49% (±1.76), 18.12% (±1.38) and 21.46% (±1.41) respectively. If we add the option of replacing infrequent words[12], error for WORD and CONST decrease to 16.51% (±1.26) and 17.71% (±1.49) respectively[13]. In Table II, one can see that the result of (PHON, default) condition have been pasted in the (PHON, INF) condition, this is because there is no need to consider infrequent words in the PHON cond.

From these results we conclude that in general conditions (without word replacement), the WORD condition (i.e. sequence of words) performs better than the two other conditions, although the improvement of performance is not striking. It is surprising that the condition using phonemes has a very close performance although it processes more complex inputs with the same reservoir size: the inputs streams are longer, and there is no cue indicating the boundaries between words. We can also notice that replacing infrequent words helps the WORD, and particularly the CONST, conditions to improve generalization. Therefore this option (of replacing infrequent words) should be used by robotic systems confronted to OOV words.

Additionally to the global hyper-parameter (HP) search, for each condition we made a HP search in order to select the optimal regularization coefficient for each condition using hyperopt with TPE (Tree of Parzen Estimators) algorithm [2] for a hundred evaluations[14]. Surprisingly PHON and WORD conditions have a similar optimal ridge parameter (2.5e-04), compared to CONST condition (5e-06). Although, from the landscape formed by the parameter exploration: WORD condition has its best values around the optimal ridge with much worse performances for other ridge values, and PHON condition has similar performances when the ridge has a lower value (i.e. less regularization). Due to this landscape, we could speculate that PHON condition is less prone to overfitting than WORD condition.

We performed additional experiments (available in supplementary materials) in order to (1) explore whether adding a feedback connection from the output units to the reservoir could improve the performance, and to (2) explore how the number of neurons inside the reservoir influence the performance of the three main conditions.

B. Tasks difficulty

In order to evaluate if each condition was of the same difficulty - in a different manner than evaluating purely generalization performance -, we did a supplementary parameter exploration. We did a TPE search changing only the number of neurons inside the reservoir without using regularization, i.e. we simply use the pseudo inverse [22] instead of ridge regression. We did this search with the infrequent word replacement option for WORD and CONST conditions. For the current data set, the optimal number of neurons are of approximately of 160, 280 and 300 neurons for the WORD, CONST and PHON conditions respectively. If we assume the optimal number of neurons (when no regularization is applied) is representative of task difficulty[15], then we could conclude that the WORD condition is easier than CONST and PHON condition. Thus it is somehow "unfair" to compare the three

[11] We successfully tested the system in a crowded noisy environment, Night of Science in Hamburg in fall 2015.

[12] i.e. occurring less than five times in the data set

[13] We outperformed previously obtained results from [17] and [18] that were applied only on CONST.

[14] The ideal ridge regularization coefficient is told to be dependent on each reservoir instance. However we prefer to use a general value a priori which is more useful for applications.

[15] Of course this is also tightly linked to overfitting problems. Another parameter search by changing the ridge actually indicates that WORD condition is much more sensitive to ridge parameter than the PHON condition.

Conditions	*Default*	*INF*	*NOISE*
PHON	**18.49 (1.76)**	$\xrightarrow{=}$ 18.49 (1.76)	33.11 (0.77)
WORD	**18.12 (1.38)**	**16.51 (1.26)**	**29.73 (0.48)**
CONST	21.46 (1.41)	17.71 (1.49)	40.53 (0.77)

TABLE II
MEAN ERROR IN PERCENT (AND STANDARD DEVIATION) FOR FULL SENTENCE COMPREHENSION FOR DIFFERENT CONDITIONS.

conditions with a number of reservoir units fixed to 500, and not allowing CONST and PHON to have more reservoir units as the tasks seem more difficult[16].

C. Noisy speech recognition

Results in Table II show that adding noisy input affect the performances[17], although they do not dramatically fall. For the PHON condition we pass from an error of 18.49% (±1.76) to 33.11% (±0.77); for WORD from 18.12% (±1.38) to 29.73% (±0.48); for CONST from 21.46% (±1.41) to 40.53% (±0.77). Interestingly the noise do not affect the three conditions in the same way. The CONST condition is the most affected by noisy inputs. This could be explained by the fact that the system could only rely on function words to parse the sentence. PHON condition is bit more affected than the WORD condition. Further explorations would be necessary to understand if this comes from the hyper-parameters that could be optimized specifically or not. Actually, the way the noisy condition is designed is "unfair" to the PHON condition, because if phonemes are used as input one would expect to have noise on the phonemes and not on the whole words.

IV. Discussion

Considering a system that learns to parse a sentence given a stream of inputs, we answered the question what could be the optimal level of abstraction of the inputs (phonemes, words or grammatical constructions) given a particular English corpus of robot commands provided by users. This optimal input abstraction may be different for other corpora, in particular if they are of different size. In this study, we only compared purely symbolic input representations and do not consider raw acoustic signals or distributed encoding representations, such as word embedding. Although we have unpublished results showing that our system is able to process words with *word2vec* representation [24] (a particular kind of embedding), we did not focus on it in this study[18].

One could argue that comparing different levels of abstraction is probably more relevant to a neuro-cognitive robot than to a child: because in order to represent words she needs to chunk phonemes, and to have constructions she should chunk groups of words and identify some words as being variables (e.g. semantic words). However, this study tries to understand *what kind of information in the inputs are most relevant for a robot to learn to parse sentences.*

In a nutshell, this study raise the question whether processing sentences as sequence of words (WORD condition) is always optimal or not. This assumption is often made by default in HRI experiments, but this may depend on the context. Our results suggest that the little advantage of a particular kind of input could change for slightly different conditions or corpora[19].

A first remark on the results is that we outperformed previous results obtained on the same corpus in CONST condition: here we obtain 17.71% (±1.49) instead of 21.4% (±2.2) in previous study [18] using the same number of reservoir units. Results showed that WORD condition (i.e. sequence of words) is the one performing best in normal conditions, but only from a short increase in performance. It remains to be explored whether these results depend on the training dataset and on its size. In particular, we speculate that the CONST condition is not performing well because the user-based corpus is small and contains a lot of sentence variants to say similar things (e.g. see Table I). Thus, the model in the CONST cond. doesn't have enough repetitions of these sentence variants in order to robustly generalize only on grammatical constructions (i.e. sentence templates). Conversely, in the WORD cond. the model can rely on the precise semantic words used in a sentence as supplementary cues to help generalization: which explains the better generalization of WORD over CONST for this small corpus.

Surprisingly the PHON cond. performs nearly as well as the WORD cond. in default experiments. The task seems more difficult in the PHON cond. because the network has to deal with longer time scales and no word segmentation is provided. We believe the PHON condition would also benefit from a bigger corpus enabling the network to extract more statistical regularities from phoneme sequences. Thus, we aim to repeat the experiment on a much larger corpus in order to demonstrate better performance in both PHON and CONST experiments.

We also explored noisy conditions, where 5% of the words were randomly replaced by other words. WORD and PHON conditions resisted better to noise than CONST condition; with WORD cond. maintaining its leadership. However, more realistic noisy conditions[20] could be obtained by randomly replace/insert/delete phonemes: we speculate that with such noise design the PHON condition would outperform the two other conditions. Consequently, we will extend this work to the processing of sequences and phonemes provided by a speech recognizer such as DeepSpeech [13]: this will provide more

[16] Effect of reservoir size is discussed in supplementary materials.

[17] All results were averaged over 50 instances instead of 100 instances.

[18] We did not focus on such representations in order to keep the comparison between levels of symbolic abstraction only. Otherwise we should also have compared it with embedding for phonemes and for grammatical constructions.

[19] We did not perform statistical comparisons between the different conditions because such tests are likely to not have enough statistical power given the size of the corpus

[20] More realistic noisy conditions could be obtained with a speech recognizer giving access to the sub-word information (i.e. phonemes recognized), instead of Google speech API for instance.

realistic phoneme/word recognition errors, and supposedly favor the PHON condition.

Despite the small corpus used, the current performances are already interesting and useful for small corpus applications in Human-Robot Interaction experiments. Because the core part of the model is a generic neural architecture, it could be easily reused or adapted for other computational or robotic experiments in language acquisition. In particular, we would like to extend this work by integrating our neural parser with multi-modal (e.g. vision, sensori-motor, ...) and behavioral robotic experiments [6]. For instance, the semantic and syntactic information of such complex sentences could be integrated into robotic experiments grounding linguistic symbols to robot behavior and to the visual modality [29], [30], [37]. Syntactic richness of natural language sentences are often simplified in such experiments (for the benefit of motor or visual modalities), and rather rely on stereotypical sequence of few semantic words without function words (e.g. "hit left blue"). Our model could help in such architectures by increase the syntactic variability a robotic architecture could deal with.

Supplementary material and source code are available at https://github.com/neuronalX/Hinaut2018_icdl-epirob

ACKNOWLEDGMENT

The author would like to thank Bhargav Teja Nallapu for valuable feedback on the paper, and Johannes Twiefel for interesting discussions.

REFERENCES

[1] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, and D. Nardi, "Effective and robust natural language understanding for human-robot interaction." in *ECAI*, 2014, pp. 57–62.

[2] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," in *Proceedings of the 12th Python in Science Conference*, 2013, pp. 13–20.

[3] H. Brouwer, M. W. Crocker, N. J. Venhuizen, and J. C. Hoeks, "A neurocomputational model of the n400 and the p600 in language processing," *Cognitive science*, vol. 41, pp. 1318–1352, 2017.

[4] F. Chang, "Symbolically speaking: A connectionist model of sentence production," *Cognitive science*, vol. 26, no. 5, pp. 609–651, 2002.

[5] P. Dominey, M. Hoen, and T. Inui, "A neurolinguistic model of grammatical construction processing," *Journal of Cognitive Neuroscience*, vol. 18, no. 12, pp. 2088–2107, 2006.

[6] P. F. Dominey and J.-D. Boucher, "Developmental stages of perception and language acquisition in a perceptually grounded robot," *Cognitive Systems Research*, vol. 6, no. 3, pp. 243–259, 2005.

[7] J. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[8] M. Eppe, S. Trott, and J. Feldman, "Exploiting deep semantics and compositionality of natural language for human-robot-interaction," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 731–738.

[9] S. L. Frank, "Strong systematicity in sentence processing by an echo state network," in *International Conference on Artificial Neural Networks*. Springer, 2006, pp. 505–514.

[10] S. L. Frank, W. F. Haselager, and I. van Rooij, "Connectionist semantic systematicity," *Cognition*, vol. 110, no. 3, pp. 358–379, 2009.

[11] S. L. Frank, P. Monaghan, and C. Tsoukala, "Neural network models of language acquisition and processing," *Preprint: stefanfrank.info*, 2017.

[12] J. Gaspers, P. Cimiano, K. Rohlfing, and B. Wrede, "Constructing a language from scratch: Combining bottom–up and top–down learning processes in a computational model of language acquisition," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 9, no. 2, pp. 183–196, 2017.

[13] A. Hannun *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[14] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[15] J. Hatori *et al.*, "Interactively picking real-world objects with unconstrained spoken language instructions," *arXiv preprint arXiv:1710.06280*, 2017.

[16] X. Hinaut and P. Dominey, "Real-time parallel processing of grammatical structure in the fronto-striatal system: a recurrent network simulation study using reservoir computing," *PLoS ONE*, vol. 8, no. 2, p. e52946, 2013.

[17] X. Hinaut, M. Petit, G. Pointeau, and P. Dominey, "Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks," *Frontiers in Neurorobotics*, vol. 8, 2014.

[18] X. Hinaut, J. Twiefel, M. Petit, P. F. Dominey, and S. Wermter, "A recurrent neural network for multiple language acquisition: Starting with english and french," in *NIPS 2015 Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*, 2015.

[19] N. Iwahashi, "Robots that learn language: Developmental approach to human-machine conversations," in *Symbol Grounding and beyond*. Springer, 2006, pp. 143–167.

[20] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," *Bonn, Germany: German National Research Center for Information Technology GMD Tech. Report*, vol. 148, p. 34, 2001.

[21] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.*, 2nd ed. Pearson International, 2009.

[22] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 659–686.

[23] R. Miikkulainen, "Subsymbolic case-role analysis of sentences with embedded clauses," *Cognitive Science*, vol. 20, no. 1, pp. 47–73, 1996.

[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. of NIPS*, 2013, pp. 3111–3119.

[25] M. Panzner, J. Gaspers, and P. Cimiano, "Learning linguistic constructions grounded in qualitative action models," in *Robot and Human Interactive Communication (RO-MAN), 2015 24th IEEE International Symposium on*. IEEE, 2015, pp. 121–127.

[26] D. K. Roy, "Learning visually grounded words and syntax for a scene description task," *Computer speech & language*, vol. 16, no. 3-4, pp. 353–385, 2002.

[27] M. Spranger and L. Steels, "Emergent functional grammar for space," *Experiments in Cultural Language Evolution*, vol. 3, pp. 207–232, 2012.

[28] L. Steels, "The synthetic modeling of language origins," *Evolution of communication*, vol. 1, no. 1, pp. 1–34, 1997.

[29] F. Stramandinoli, D. Marocco, and A. Cangelosi, "The grounding of higher order concepts in action and language: a cognitive robotics model," *Neural Networks*, vol. 32, pp. 165–173, 2012.

[30] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes," *Adaptive behavior*, vol. 13, no. 1, pp. 33–52, 2005.

[31] T. Taniguchi, T. Nagai, T. Nakamura, N. Iwahashi, T. Ogata, and H. Asoh, "Symbol emergence in robotics: a survey," *Advanced Robotics*, vol. 30, no. 11-12, pp. 706–728, 2016.

[32] M. Tenorth and M. Beetz, "Knowrob: A knowledge processing infrastructure for cognition-enabled robots," *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013.

[33] J. Twiefel, T. Baumann, S. Heinrich, and S. Wermter, "Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing," in *Twenty-Eighth AAAI. Québec City, Canada*, 2014, pp. 1529–1535.

[34] J. Twiefel, X. Hinaut, M. Borghetti, E. Strahl, and S. Wermter, "Using Natural Language Feedback in a Neuro-inspired Integrated Multimodal Robotic Architecture," in *Proc. of RO-MAN*, New York City, USA, 2016.

[35] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[36] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences." *Ph. D. thesis. Harvard University, Cambridge, MA.*, 1974.

[37] T. Yamada, S. Murata, H. Arie, and T. Ogata, "Dynamical integration of language and behavior in a recurrent neural network for human–robot interaction," *Frontiers in neurorobotics*, vol. 10, p. 5, 2016.