Learning to Parse Grounded Language using Reservoir Computing

Xavier Hinaut, Michael Spranger

Learning to Parse Grounded Language using Reservoir Computing

1st Xavier Hinaut
Inria
LaBRI, UMR 5800, Bordeaux INP
IMN, UMR 5293, Université de Bordeaux
Bordeaux, France
xavier.hinaut@inria.fr

2nd Michael Spranger
Sony Computer Science Laboratories
Tokyo, Japan
michael.spranger@gmail.com

***Abstract*—Recently new models for language processing and learning using Reservoir Computing have been popular. However, these models are typically not grounded in sensorimotor systems and robots. In this paper, we develop a model of Reservoir Computing called Reservoir Parser (ResPars) for learning to parse Natural Language from grounded data coming from humanoid robots. Previous work showed that ResPars is able to do syntactic generalization over different sentences (surface structure) with the same meaning (deep structure). We argue that such ability is key to guide linguistic generalization in a grounded architecture. We show that ResPars is able to generalize on grounded compositional semantics by combining it with Incremental Recruitment Language (IRL). Additionally, we show that ResPars is able to learn to generalize on the same sentences, but not processed word by word, but as an unsegmented sequence of phonemes. This ability enables the architecture to not rely only on the words recognized by a speech recognizer, but to process the sub-word level directly. We additionally test the model's robustness to word error recognition.**

KEYWORDS

Language acquisition; Grounding of Knowledge and Development of Representations; Language and semantic reasoning

INTRODUCTION

Language learning in grounded, developmental settings is a hugely interesting but daunting task [1]. Many researchers have tackled different parts of this topic studying phoneme acquisition [2], word grounding [3] and cross-situational learning [4]. All of these experiments and results focus on important subtasks of the problem.

One key sub problem is how to learn not just the meaning of individual words but the meaning of compositional structures, e.g. sentences and phrases. From the beginning of the field compositionality has been one of the key targets for research because it is one of the key features of human language. There are two key questions related to the problem of compositionality. One is how to represent grounded compositional semantics. Recent proposals include representing semantics with grounded procedural structures in a system called Incremental Recruitment Language (IRL) [5]. These proposals argue that we best understand compositional structure in terms of conceptualization operations that serve different functions such as identifying a referent, describing an event, commanding other agents. But these functions must be grounded in processing of sensorimotor spaces and in general cognitive abilities such as categorization, mental rotation etc.

Another key problem in language learning is how to learn the mapping between language syntax and compositional semantics. For this problem, we have seen the application of various approaches, such as, grounded symbolic methods [6] (for grounded learning with IRL), early recurrent neural networks [7] and also recently deep learning [8]. One of the interesting approaches that has come out of this line of work is Reservoir Computing-based models, and in particular Echo State Networks (ESN) [9]–[11]. Such approaches are often more inspired by brain mechanisms involved in language processing [10], [12]. Recently, these models have shown great potential in syntactic generalization [12]–[14] – that is the processing of sentences with complex structures without using word semantics. However, especially the latter brain-inspired models have not been applied to grounded language learning settings yet.

In this paper, we combine earlier work on compositional semantics using Incremental Recruitment Language (IRL) with an Echo State Network-based parser (ResPars) in a grounded architecture. We show that ResPars is able to learn from developmental experiments with grounded data. We start by describing related work, followed by the grounded setup. We then describe the two key systems IRL and ResPars. We conclude with experiments and results.

RELATED WORK

Grounded language learning has often focused on the acquisition of categories and concepts [15], for example in the spatial language [16] and color domains [17]. In that work, centroid-based classifiers with distance (Voronoi tessellation) or other Machine Learning algorithms (neural networks, Bayesian models etc.) are used for representing grounded word meanings. The work presented in this paper differs from those approaches by focusing on grammar and compositional semantics.

Fewer models learn syntax in grounded settings. One research strand has explored how to learn symbolic grammars in the framework of *language games*. Models have been

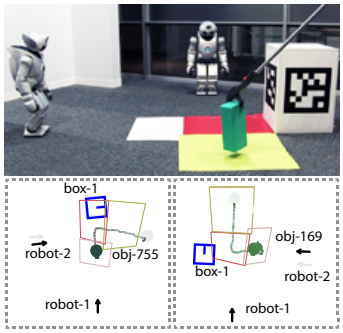

Fig. 1. Dynamic movement setup. Left: world model extracted by the left robot. Right: world model computed by the right robot. Estimated movement of the block (circle) is visualized through opacity. The starting point has a lower opacity (alpha). Regions are visualized by colored quadrangles. The blue square shows position, orientation of the box. Arrows are robots. `robot-1` is the origin of the coordinate system, `robot-2` position and orientation of the other robot. Figure adapted from [26].

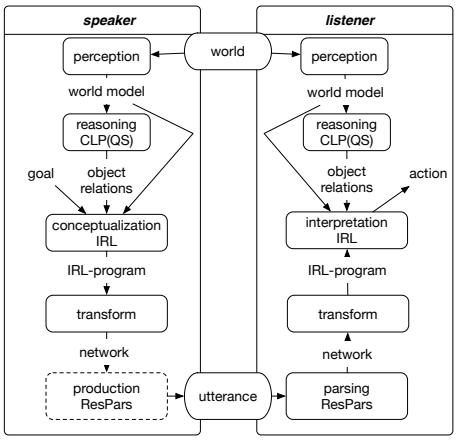

Fig. 2. Systems involved in enabling tutor and learner to communicate. (Notice that ResPars for production is future work.)

developed in particular for spatial language [6], [18], temporal language learning [19] and events [20]. A lot of this research focuses on framing language learning as learning constructions; it is often motivated by insights from Cognitive Linguistics, and in particular Cognitive Semantics, Construction Grammar and Usage-based linguistics. The usage-based approach to language learning has also led to studies examining specific aspects of language such as ditransitive constructions and others [21]. All these models show that some constrained set of constructions can be fruitfully learnt using relatively simple learning operators that manipulate symbolic structures. Some of these models also learn directly on sensorimotor data such as [18]. The approach presented in this paper reuses some of this work for modeling semantics but replaces the symbolic grammars with brain-inspired learning mechanisms and representations.

The approach discussed in this paper is close to a field called *semantic parsing* in Natural Language Processing. This field is recently dominated by deep learning approaches [22]. For instance, since Mikolov's work on Word2Vec [23] which showed that deep learning methods can provide useful representations for semantic operations such as *queen* = *king* - *man* + *woman*, much effort has been put into word embedding [24] and sentence representation [25]. Typically these methods require huge data sets for training and are only based on the statistics of corpora, not on any grounded reality. Thus they are not good candidates for grounded developmental approaches.

System Architecture and Setup

We study grounded language acquisition using robots that interact in an office space. The basic setup is the following. There are two or more robots drawn from a larger population: one is the tutor, the other is the learner. Both robots try to draw each others attention to objects in the environment using natural language. For this paper we combine two setups: one about locations of objects and one about dynamic movements (Figure 1).

The environment consists of a number of blocks, boxes and interlocutors. For the dynamic movement setup, we also include colored regions on the floor. The vision system of each robot tracks objects and establishes a model of the environment with real-valued distances and orientations of objects with respect to the body of the robot. The environment is open-ended. Objects and boxes are added or removed and their spatial configuration changed. Moreover, robots are free to move around. Objects change positions in the dynamic setup over time. We record these positions not just at a specific moment in time but over the whole episode.

Traditionally in *language game* studies, tutor and learner engage in scripted interactions. However, for this paper we are focussing on parsing. Figure 2 details the overall system architecture and subsystems involved in producing and interpreting utterances in our system. In particular there are two systems that are important for this paper. Incremental Recruitment Language (IRL) and the Reservoir Computing based parser (ResPars). IRL is a framework for representing and learning categories and compositional semantics. It bridges between real-valued, continuous sensorimotor representations and symbolic meaning. ResPars is a construction grammar inspired neural network approach allowing to express mappings between symbolic meanings and Natural language sequences (words and grammar, as well as phonemes) and back.

Grounded Semantics with IRL

We use *Incremental Recruitment Language* (IRL) [5] to represent the semantics of Natural language phrases. IRL is a *procedural* meaning framework that stresses the importance of conceptualization processes over truth-values for Natural language semantics. IRL represents the meaning of utterances as constraint programs. For this paper we combine previous

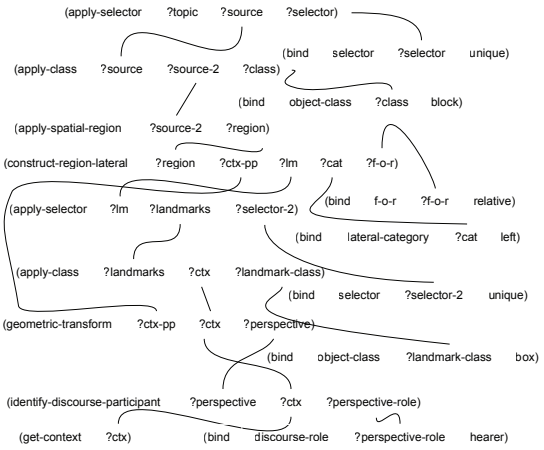

Fig. 3. Semantic structure for "The block left of the box from your perspective".

work on color word semantics [17], modeling locative spatial semantics [27] and dynamic movement events [26].

Figure 3 shows the IRL-program (meaning) of the phrase "The block left of the box from your perspective". The structure contains pointers to categories and spatial relations in the form of *bind statements*, as well as a number of *cognitive operations*. For example, `construct-region-lateral` constructs a spatial similarity region based on a lateral category (such as left). Cognitive operations and bind statements are linked using variables (which are symbols starting with ?).

We implemented different types of categories and relations: selectors, concepts, color categories, spatial relations, dynamic spatial relations and movement events

Object categories: Objects in our scenarios can be categorized: robots, boxes, blocks and regions.

Color Categories: We use here 3 color categories: "yellow, "green", "red". We represent color categories using a similarity function based on a prototypical color [17].

$$\mathrm{sim}_c(o, c) \quad := \quad e^{-\frac{1}{2\sigma_c}|o_o - c_c|}$$

where o is some object, c the category, o_c the color of a particular object o and c_c the prototypical color of category c.

Selectors: In this paper we focus only on the determiner "the", which is modeled as requiring that the object is unique. In practice that means that there needs to be an object with a much higher similarity than other objects. That object then becomes the referent of the whole (sub) IRL-program.

Spatial relations: We implemented *projective categories* such as "front", "back", "left" and "right". Similar to color categories, these categories are represented using similarity functions.

Dynamic spatial relations: More complicated dynamic categories such as *into*, *out-of* and *across* are implemented using a spatial reasoner (more detail are in [26], [28]).

Cognitive Operations: Agents can use categories and relations and combine them with different cognitive operations into IRL-programs. For example, *categorization operations* take a set as input and score objects according to some similarity functions defined by categories. Examples are `apply-class` and `apply-category`. *Mental rotations* are implemented as linear algebra operations that transform a feature space such as angle and direction to another point of origin, e.g. `geometric-transform`. These operations also handle different frames of reference (intrinsic, relative and absolute).

RESPARS: THE RESERVOIR PARSER MODEL

ResPars proposes to model how the human brain processes sentences and is inspired from several studies in neuroscience [10], [12]. A schema of the global architecture can be seen on Figure 4. 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 sub-part of the basal ganglia). Both are generally involved in sequence processing and learning, and in particular in sentence parsing.

The task of ResPars is to do semantic role labelling, which is equivalent to finding all the correct roles of the content words of a sentence. In other words, the goal of ResPars is to learn the mapping of the semantic words of a sentence (i.e. content words like nouns, verbs, adjectives, adverbs) onto different slots (the semantic roles: e.g. action, location) of a basic event structure (e.g. *action(object, location)*). In this study, ResPars has been adapted (compared to previous studies) to work jointly with IRL: (1) it processes all words (content and function words) indifferently, and not only the sentence structure (i.e. *grammatical constructions* [29]) like previously; (2) it represents the outputs as a graph adjacency matrix which maps content words with their roles and all their potential links with one another. The task of ResPars is similar to semantic parsing but with fewer roles: only relevant roles for the robot experiment have been kept. The output of ResPars is thus a graph matrix representation: bipartite graph linking words to meanings and roles. For learning purposes this is represented as a big vector which is a concatenation of the matrix rows. The new output graph representation enables to have a representation that is independent on the presence of optional arguments (e.g. adjectives for nouns[1]).

In Figure 4 we can see that sentences are given as sequences of words or phonemes[2] depending on the experimental condition. Then the dynamics of the reservoir are trained to be associated with the output layer (i.e. the read-out layer). As we said previously, the output layer consists of the matrix defining the graph linking each word (by its order in the sentence) to their roles and their link to other words. For the sentence given as example in Figure 4 the word *put* has the role of predicate and is linked to *toy* and *left*.

[1]If a noun has no adjective its node will have no links to other words.
[2]Not both (word and phonemes) at the same time.

ECHO STATE NETWORKS

ResPars is based on Echo-State-Network – in short: ESN [9] – 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. The random weights of the ESN's reservoir are scaled to possess suitable dynamics (e.g. "edge of chaos"). The objective is to have reservoir states that are linearly separable and that can be mapped to the output layer using a computationally cheap linear regression. 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, we use ridge regression (also known as regression with Tikhonov regularization), which probably provides the most stable solution in this context [30]:

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

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 $\mathbf{M}$.

EXPERIMENTS

Listener: ResPars to IRL: For each utterance, the sentence is first processed by ResPars. In both experimental conditions, the words or phonemes are given one after another: at each time step the reservoir receives a given word/phoneme. The stream of word/phoneme inputs activates the reservoir (i.e. the random recurrent network) and creates particular dynamics for each sentence in high-dimensional space. The output is then trained to represent the sentence meaning as a graph that will be given as input to the IRL framework. This output representation enables to transform the graph as an IRL-program which can then interpreted and linked with grounded world model and reasoning modules (see Figure 2). The processing of sentences is sequential (one word/phoneme at a time) and the final estimation of the thematic roles for each SW (i.e. semantic word) is read-out at the end of the sentence. A current estimation of the predicted roles is however available at each time step.

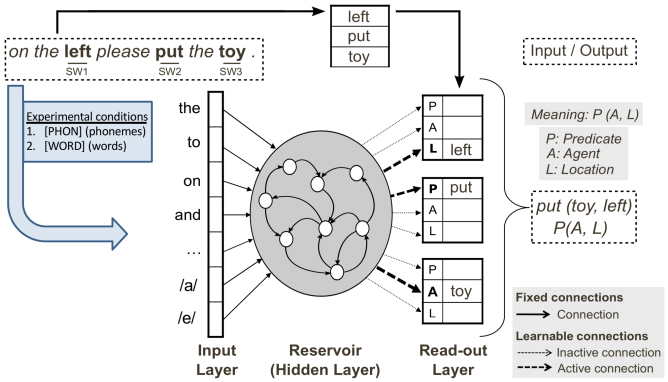

Fig. 4. **ResPars model with two different input conditions: words and phonemes.** The system processes inputs as follows: (top left) from a sentence as input, the model outputs (middle right) the roles for each semantic word (SW). The processing of the sentence is sequential: each phoneme or word is given one at a time. On the figure we only represent relevant outputs for the simple example given (and not the full graph adjacency matrix), because several words can have empty roles. Figure adapted from [31].

Training of ResPars: To train and test the ResPars parser we do a 10-fold cross-validation: we split the data in 10 folds and take nine folds to train the reservoir, and the last fold to test the generalization. We do this cross-validation for 10 different ResPars networks (which have different random weights) and average the results. We used offline ridge regression: all the nine folds are used in one batch. The hyper-parameter used for the experiment were not optimized for this particular experiment but taken from previous work: spectra radius = 1, input scaling = 0.6, Wstd = 0.1 and leak rate α = 0.06. The only parameter changed is the regularization coefficient: $2.5 * 10^{-6}$.

RESULTS

Corpus

From the grounded data, we can generate a corpus of IRL-programs that have been used or could be used by robots to communicate in the real world. We identified 19 different semantic IRL-program patterns representing the meaning of simple noun-phrase type semantics up to semantics of phrases such as "The block left of the box from your perspective" and the "The block moves across the left red region". Combined with lexical material such as color categories, object classes, dynamic spatial relations etc, we generated a corpus of 2378 IRL-programs with corresponding phrases for the ResPars to train on. Some sentences generated had several adjectives per noun (up to three per noun), which makes the task harder for ResPars because it changes a lot the output structure even if the meaning of the sentence does not change much.

ResPars generalization (words)

Figure 5 shows that ResPars model is able to learn the corpus well. It reaches a very good performance: it is making

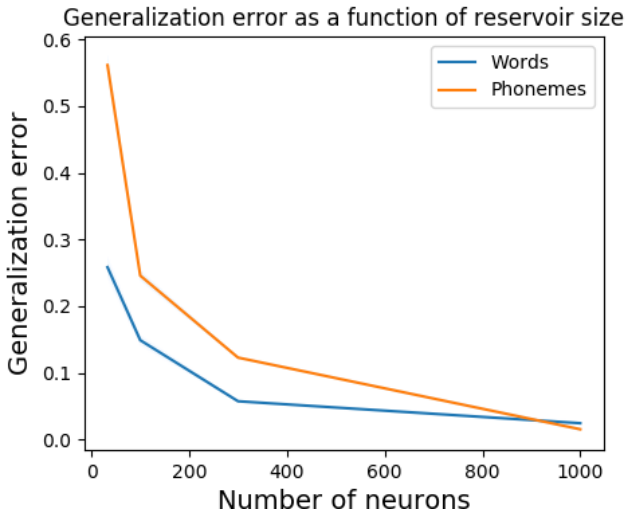

Fig. 5. Generalization error of ResPars depending on the size of the reservoir, for word and phoneme conditions.

less than 3% of error on test sentences (not seen in training data) with 1000 neurons in the reservoir. The error is expressing the number of sentences for which at least one role is incorrect. Thus no error means that all the roles for sentences are correctly retrieved. This is a hard measure of performance.

The generalization errors (best is 0, worst is 1) obtained as a function of the number of neurons (N) in the reservoir are the following: N33: 0.258(+/-0.0235); N100: 0.149(+/-0.0115); N300: 0.0572(+/-0.0033); N1000: 0.0242(+/-0.0044).

The number of neurons required to learn and generalize is also small. ResPars of sizes as small as 30 neurons could already generalize quite well.

ResPars generalization (phonemes)

As the model is working well on parsing sentences, we performed a more difficult experiment by replacing the words by the sequence of phonemes (in a similar manner as [31]). The generalization errors obtained in function of the number of neurons (N) in the reservoir are the following: N33: 0.5616 (+/-0.0934); N100: 0.2457 (+/-0.0170); N300: 0.1227 (+/-0.0116); N1000: 0.0151 (+/-0.0031).

Comparison of conditions

As expected the performance for phonemes condition is lower than for word condition for small reservoirs, but the results are still good for such a task of parsing relying on phonemes instead of words. For instance, the performance of phoneme-ResPars with 300 neurons is comparable (and slightly better) than the performance of word-ResPars with 100 neurons.

A surprising result is that the performance for a reservoir of 1000 neurons is better for the phoneme condition (1.51% error) than for the word condition (2.42% error). It is unexpected, because the task is more difficult in the phoneme condition: the stream of inputs is much longer[3]. Thus the reservoir needs to keep more information in working memory to succeed in the task: this is why we observe such errors rates for small reservoirs: e.g. errors for phoneme-N300 condition are comparable with word-N100 condition. The difference on word-based and phoneme-based could not be stated significantly, both errors are near optimal performance and more experiments would be needed to show that one condition is better (in general) than the other: varying the corpus, the number of neurons or hyper-parameters of the model. In other words, the take-home message is not that one condition (word vs. phoneme) is better than the other, but that the ResPars model is able to perform well with both approaches; thereby demonstrating its robustness. Such robustness on phoneme/word levels, along side with the relative robustness to word replacement, are useful to overcome speech-to-text recognition errors.

These good performances with phonemes means that it adds the possibility to not rely on the word speech recognition system while run on a robot. ResPars could directly rely on the stream of phonemes instead of the stream of words given by the speech recognition. This is an interesting potential for the IRL-ResPars setup: this means that the experiments could be performed without word knowledge prerequisite, which is an interesting perspective for developmental experiments of grounded language.

Robustness to noise

We made a supplementary experiment to investigate whether ResPars was robust to noise and sensible to errors in word recognition. Speech recognition systems have made a lot of progress the past decade, but the best systems still have around 5% WER (Word Error Rate): this is non negligible when considering the impact of the recognition on a full sentence. Twiefel et al. [32], [33] previously showed that even when using Google speech recognition API for Human-Robot Interaction, one has to post-processing the results by constraining to plausible vocabulary, otherwise too few sentences are correctly parsed to create a good interaction.

For the word condition, we performed four experiments with varying the probability of a word being "misrecognized": i.e. the word was randomly replaced by another word of the corpus. Note that such random replacement of words is a difficult task, because a noun or an adjective could be replaced by a function word like "of", "in", "my", "into": which changes the grammatical structure of the sentence and often makes it impossible to understand.

We did experiments with four different levels of noise: 1%, 5% and 10% probability of a word being randomly replaced. The following results were obtained: 0.0900 (+/-0.0013) for 1%, 0.1860 (+/-0.0022) for 5% 0.2676 (+/-0.0024) for 10% and 0.3643(+/-0.0032) for 20% noise. ResPars seems

[3] n time longer, with n the average number of phonemes per words)

quite robust for this difficult task: for 5% of word randomly replaced, it is still able to parse correctly 81.40% of the sentences, which makes it a robust model for online Human-Robot or Robot-Robot Interactions.

The corpus and code will be available on Github: https://github.com/neuronalX/EchoRob.

Discussion

Further work would extend this work to include a sentence production version of ResPars: in order to produce a sentence given an IRL-program. To achieve social grounding, sentence parsing is not enough but the learner must also speak: it enables the close the sensori-motor (parsing-production) loop. Parsing is constrained by the world and possible interpretations. Missing information in parsing can in some cases be provided by the world, context and grounding. However, production is less constrained and choices in production have a big impact on whether or not a phrase can be understood.

ResPars grounds language in semantic structure (IRL-programs). However, that presumes that the set of IRL-programs is already known to the learner. That is of course, a simplifying assumption and learners actually should learn parsing (syntax to semantics) and interpretation (semantics to pragmatics and grounding) together. There has been earlier work achieving co-acquisition of syntax and semantics for symbolic grammars (with IRL), but to the best of our knowledge there are no other brain-inspired architecture models doing so yet.

Another direction for extending the current work could extend the ResPars model to *Conceptors* [34]: they are natural extension of reservoirs which are able to learn sequential patterns in order to easily interpolate and make logical operations on them. Conceptors are able to generalize by interpolation (like neural networks in general) but also to generalize by extrapolation (outside the domain of training examples) which is less common. This would allow to share the compositionality of IRL to the Conceptor of ResPars, for example in order to represent abstract actions [35] composed of several sub-actions. In this way, ResPars could include abstract semantics of action verbs through this important feature of compositionality.

References

[1] A. Cangelosi et al., "Integration of action and language knowledge: A roadmap for developmental robotics," *Autonomous Mental Development, IEEE Transactions on*, 2010.

[2] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *Evolutionary Computation, IEEE Transactions on*, 2007.

[3] F. Stramandinoli, D. Marocco, and A. Cangelosi, "Making sense of words: a robotic model for language abstraction," *Autonomous Robots*, 2017.

[4] P. F. Dominey and J.-D. Boucher, "Developmental stages of perception and language acquisition in a perceptually grounded robot," *Cognitive Systems Research*, 2005.

[5] M. Spranger, S. Pauw, M. Loetzsch, and L. Steels, "Open-ended Procedural Semantics," in *Language Grounding in Robots*. Springer, 2012.

[6] M. Spranger, "Incremental grounded language learning in robot-robot interactions - examples from spatial language," in *2015 ICDL-Epirob*. IEEE, 2015.

[7] Y. Sugita and J. Tani, "Learning semantic combinatoriality from the interaction between linguistic and behavioral processes," *Adaptive behavior*, 2005.

[8] 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*, 2016.

[9] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," *GMD Tech. Report*, 2001.

[10] P. Dominey, M. Hoen, and T. Inui, "A neurolinguistic model of grammatical construction processing," *Journal of Cognitive Neuroscience*, 2006.

[11] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," in *Advances in neural information processing systems*, 2010.

[12] 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*, 2013.

[13] X. Hinaut et al., "Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks," *Frontiers in Neurorobotics*, 2014.

[14] ——, "Corticostriatal response selection in sentence production: Insights from neural network simulation with reservoir computing," *Brain and language*, 2015.

[15] M. Spranger and K. Beuls, "Referential uncertainty and word learning in high-dimensional, continuous meaning spaces," in *ICDL-Epirob*, 2016.

[16] M. Spranger, "Grounded lexicon acquisition - case studies in spatial language," in *ICDL-Epirob 2013*. IEEE, 2013.

[17] J. Bleys, M. Loetzsch, M. Spranger, and L. Steels, "The Grounded Color Naming Game," in *Proceedings of the 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009.

[18] M. Spranger and L. Steels, "Co-acquisition of syntax and semantics - an investigation in spatial language," in *IJCAI 2015*. AAAI Press, 2015.

[19] K. Gerasymova and M. Spranger, "An experiment in temporal language learning," in *Language Grounding in Robots*. Springer, 2012.

[20] L. Steels, M. Spranger, R. van Trijp, S. Höfer, and M. Hild, "Emergent action language on real robots," in *Language Grounding in Robots*. Springer, 2012.

[21] N. C. Chang and T. V. Maia, "Learning grammatical constructions," in *Proc. 23rd Cognitive Science Society Conference*, 2001.

[22] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP 2013*, 2013.

[23] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *NAACL 2013*, 2013.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[25] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," *arXiv:1602.03483*, 2016.

[26] M. Spranger, J. Suchan, and M. Bhatt, "Robust natural language processing — combining reasoning, cognitive semantics, and construction grammar for spatial language," in *IJCAI 2016*. AAAI Press, 2016.

[27] M. Spranger, *The Evolution of Grounded Spatial Language*. Language Science Press, 2016.

[28] M. Bhatt, J. H. Lee, and C. Schultz, "Clp (qs): a declarative spatial reasoning framework," in *International Conference on Spatial Information Theory*. Springer, 2011.

[29] A. Goldberg, *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, 1995.

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

[31] X. Hinaut, "Which input abstraction is better for a robot syntax acquisition model? phonemes, words or grammatical constructions?" in *2018 ICDL-Epirob*. IEEE, 2018.

[32] J. Twiefel, T. Baumann, S. Heinrich, and S. Wermter, "Improving domain-independent cloud-based speech recognition with domain-dependent phonetic post-processing," in *23rd AAAI*, 2014.

[33] J. Twiefel, X. Hinaut, M. Borghetti, E. Strahl, and S. Wermter, "Using natural language feedback in a neuro-inspired integrated multimodal robotic architecture," in *RO-MAN*. IEEE, 2016.

[34] H. Jaeger, "Controlling recurrent neural networks by conceptors," *arXiv preprint arXiv:1403.3369*, 2014.

[35] F. Stramandinoli, D. Marocco, and A. Cangelosi, "The grounding of higher order concepts in action and language: a cognitive robotics model," *Neural Networks*, 2012.