Cross-Situational Learning with Reservoir Computing for Language Acquisition Modelling

Alexis Juven, Xavier Hinaut

Cross-Situational Learning with Reservoir Computing for Language Acquisition Modelling

Alexis Juven
1. INRIA Bordeaux Sud-Ouest.
2. LaBRI, Bordeaux INP, CNRS, UMR 5800.
3. Institut des Maladies Neurodégénératives,
Université de Bordeaux, CNRS, UMR 5293.
Bordeaux, France.

Xavier Hinaut
1. INRIA Bordeaux Sud-Ouest.
2. LaBRI, Bordeaux INP, CNRS, UMR 5800.
3. Institut des Maladies Neurodégénératives,
Université de Bordeaux, CNRS, UMR 5293.
Bordeaux, France.
orcid.org/0000-0002-1924-1184

***Abstract*—Understanding the mechanisms enabling children to learn rapidly word-to-meaning mapping through cross-situational learning in uncertain conditions is still a matter of debate. In particular, many models simply look at the word level, and not at the full sentence comprehension level. We present a model of language acquisition, applying cross-situational learning on Recurrent Neural Networks with the *Reservoir Computing* paradigm. Using the co-occurrences between words and visual perceptions, the model learns to ground a complex sentence, describing a scene involving different objects, into a perceptual representation space. The model processes sentences describing scenes it perceives simultaneously via a simulated vision module: sentences are inputs and simulated vision are target outputs of the RNN. Evaluations of the model show its capacity to extract the semantics of virtually hundred of thousands possible combinations of sentences (based on a context-free grammar); remarkably the model generalises only after a few hundred of partially described scenes via cross-situational learning. Furthermore, it handles polysemous and synonymous words, and deals with complex sentences where word order is crucial for understanding. Finally, further improvements of the model are discussed in order to reach proper reinforced and self-supervised learning schemes, with the goal to enable robots to acquire and ground language by them-selves (with no oracle supervision).**

***Index Terms*—Recurrent Neural Networks, Reservoir Computing, Echo State Networks, Language Learning, Cross-situational Learning, Unsupervised Learning, Language Acquisition.**

I. Introduction

Language unfolds in time. One difficulty when processing natural language with neural networks architectures is to manage this unfolding in time. Neural systems have to deal with the sequence of words if they want to extract precisely the meaning of sentences (e.g. instead of taking a bag-of-words approach). They can either use feed-forward networks and encode time as a spatial property (by unfolding the whole sentence into one input vector), or use recurrent neural networks (RNNs) to parse one word at a time. Recently, RNNs, in particular LSTMs [1] or GRUs [2] and their variants [3], were successfully trained in a supervised fashion on a variety of tasks: predict the next element in a sequence, timeseries classification, sequence to sequence mapping [4]. What is impressive is the rich dynamic latent space that appears through supervised training. For instance, some units specialized in detecting the opening and closing of quotes or brackets: this is obtained "only" by training a LSTM to predict the next character in a very large corpus [5].

However, acquiring language is not a supervised task: e.g. before one year of age, children are able to segment words from speech based on statistical learning mechanisms [6]. Moreover, language acquisition is not only a matter of processing or producing streams of sounds, the child has to learn to map these sounds to multi-modal features and dynamics of the world: e.g. learning that the sound stream /'æpəl/ is referring to a green or red fruit (i.e. *apple*).

Symbolic algorithms have been proposed to explain how children could learn word-to-meaning mappings [7]. Smith et al. [8] showed that children quickly learn word-referent mappings via cross-situational statistics. The idea is the following: infants listen to utterances that describe one object, but they do not know for sure which is the object of focus because many objects are in their field of view; however, through repeated presentation of the same object with the same spoken word, they are able to learn word-referent mapping based on co-occurrences of the word with the object.

Several models of cross-situational learning have been proposed: some of them used social cues to help the learning of particular words [9]. Some robotic experiments modelled the grounding of synonyms in cross-situational learning [10]. Taniguchi et al. [11] propose a virtual robotic framework to learn to ground words from vision features based of short sentences void from function words: e.g. "grasp green right ball". However, to our knowledge there is no architecture that tries to model how children could learn to understand directly from full sentences through cross-situational learning, without providing specific cues (e.g. social or prosodic).

Previously, we proposed a model called *ResPars* (for Reservoir Parser) to model how the brain processes complex sentences with no prior knowledge on the words. In this study, we want to demonstrate that this model could go one step further towards biologically and developmentally plausible learning schemes. Namely, that it is not necessary to train it with accurate supervised signals, but based on its capacity to extract statistics form corpora [12] it could also be trained in cross-situational learning conditions, i.e. with noisy teacher signal.

Moreover, the model should have feedback only at the end of sentences and be trained fully online [13]. Finally, we want to test its ability to directly output concepts (such as the object, its color and the position of the object) instead of the thematic roles of words.

II. Methods

Python implementation of the presented model is available on https://github.com/aJuvenn/JuvenHinaut2020_IJCNN

A. Reservoir Computing

Reservoir Computing (RC) is an efficient paradigm to train RNNs by keeping an important part of weights untrained (i.e. random). In this study we used Echo State Networks (ESNs) [14], which are an instance of the RC paradigm [15]. Figure 1 illustrates how an ESN works.

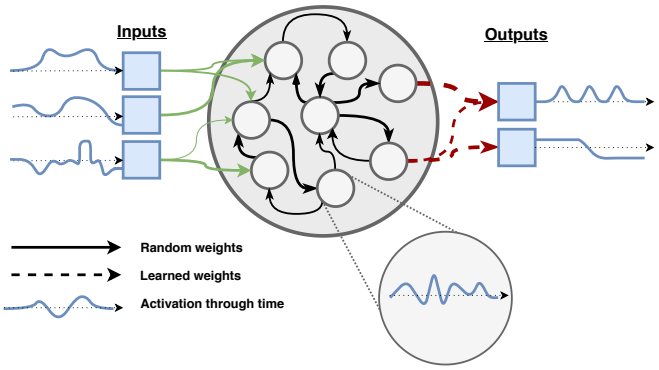

Fig. 1. Reservoir Computing. Echo State Networks are an instance of the Reservoir Computing paradigm using units with continuous states (as opposed to spiking activity). Coherent outputs are generated from the chaotic internal activity.

ESN 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 criticality"). 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. The equations 1 and 2 define the update of the ESN receiving input $\mathbf{x}_t$ and producing output $\mathbf{y}_t$.

$$\mathbf{r}_t \leftarrow (1-\alpha)\ \mathbf{r}_{t-1}\ +\ \alpha\ \mathbf{tanh}(\mathbf{W}^{\text{rec}}\mathbf{r}_{t-1} + \mathbf{W}^{\text{in}}\mathbf{x}_t) \quad (1)$$

$$\mathbf{s}_t \leftarrow \begin{pmatrix} 1 \\ \mathbf{x}_t \\ \mathbf{r}_t \end{pmatrix} \qquad \mathbf{y}_t \leftarrow \mathbf{W}^{\text{out}}\mathbf{s}_t \quad (2)$$

where $\mathbf{r}_t$ and $\mathbf{s}_t$ are the recurrent layer firing rate and the network internal state respectively at time t. $\mathbf{W}^{\text{in}}$, $\mathbf{W}^{\text{rec}}$ and $\mathbf{W}^{\text{out}}$ are the input, the reservoir, and the read-out matrices respectively.

In the case of an offline learning method such as ridge regression, the training dataset $(\mathbf{x}_t, \mathbf{z}_t)_{t\in[1;d]}$, where $\mathbf{z}_t$ is the expected output after reading input $\mathbf{x}_t$, must be known in advance. The output layer $\mathbf{W}^{\text{out}}$ is created according to equation 3 by feeding the network with training inputs.

$$\mathbf{W}^{\text{out}} \leftarrow \mathbf{Z}\mathbf{S}^\top(\mathbf{S}\mathbf{S}^\top + \varepsilon\mathbf{I})^+ \quad (3)$$

where $\mathbf{S} = \begin{pmatrix}\mathbf{s}_1 & \cdots & \mathbf{s}_d\end{pmatrix}$ is a matrix made of the different internal states $\mathbf{s}_t$ produced after reading $\mathbf{x}_t$ during the learning phase, $\mathbf{Z} = \begin{pmatrix}\mathbf{z}_1 & \cdots & \mathbf{z}_d\end{pmatrix}$ the matrix of their corresponding expected network outputs, and ε a regularisation term. $\mathbf{A}^+$ notation refers to the Moore–Penrose inverse matrix of $\mathbf{A}$.

B. Training reservoir with FORCE learning

In order to update the output weights $\mathbf{W}^{out}$ after each learning example, an online method called FORCE learning [16] is used. This method relies on a matrix $\mathbf{P}$ which is saved and updated after each learning step in order to have a reminder of the previous encountered contexts, allowing to compute an approximation of a pseudo-inverse matrix. At step t, if the network produces the output $\mathbf{y}_t$ after producing an internal state $\mathbf{s}_t$, the equations 4 and 5 are used to make the network learn the expected output $\mathbf{z}_t$ by incrementing $\mathbf{W}^{out}$ of $\delta\mathbf{W}_t$.

$$\mathbf{P}_0 \leftarrow \frac{\mathbf{I}}{\varepsilon} \qquad \delta\mathbf{P}_t \leftarrow -\frac{\mathbf{P}_{t-1}\mathbf{s}_t\mathbf{s}_t^\top\mathbf{P}_{t-1}}{1+\mathbf{s}_t^\top\mathbf{P}_{t-1}\mathbf{s}_t} \quad (4)$$

$$\mathbf{W}_0^{out} \leftarrow \mathbf{0} \qquad \delta\mathbf{W}_t \leftarrow (\mathbf{z}_t - \mathbf{y}_t)\mathbf{s}_t^\top\mathbf{P}_t \quad (5)$$

Matrix $\mathbf{W}^{out}$ update is similar to a gradient descent method, but using matrix $\mathbf{P}$ as an adaptive learning rate, parameterized with a regularisation term ε. This allows fast and stable learning, which is useful in case of noisy learning examples.

C. ResPars: The Reservoir Parser Model

ResPars proposes to model how the human brain processes sentences and is inspired from several studies in neuroscience [12], [17]. 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). It was used also use in cognitive robotics in applications of Human-Robot Interaction (HRI) [18]–[21]. In upper left part of Figure 2 we can see that sentences are given as sequences of words. Then the dynamics of the reservoir are trained to be associated with the output layer (i.e. the read-out layer).

The task of previous versions of ResPars was 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 was 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)*). ResPars has been also adapted to work jointly with IRL [22], a grounded multi-agent robotic framework [23], for

which outputs were adapted to represent a graph adjacency matrix which maps content words with their roles and all their potential links with one another.

D. Towards cross-situational learning

In this study we adapt ResPars model to cross-situational learning in order to make it work in more realistic conditions towards language acquisition. The model together with the learning procedure is schematized in Figure 2: the reservoir takes a sentence as input and returns as output a perceptual representation of what the sentence is describing. Output of the network are visual concepts allowing it to represent one or several objects with different characteristics (e.g. category, position, color). These concepts are activated after hearing a sentence describing a scene: the model is able to provide which objects are described in the sentence and their characteristics.

The network learns with an efficient online algorithm, using cross-situational learning. It learns by observing what a vision module provides him while it hears a sentence describing totally or partially the scene it is observing. The vision module output has the same encoding as the network output: a vector of zeros and ones which describes whether a characteristic is present or not. The network hears a sentence, and learns to reproduce, from its final internal state, what the model sees at the same time. This means that the teacher output is not exactly what is expected, since the sentence may not mention everything that is seen. We expect that statistically, by repeated co-occurrence of the words (embedded in the sentence flow) together with the perceived concepts, only the pertinent concept will be kept activated when hearing a sentence. We hypothesise that given the quick generalisation ability of reservoirs and the fast learning algorithm used (i.e. FORCE), we should obtain correct scene perception after few learning steps implying the same objects.

To compare this model to the previous ResPars one, this one performs online learning with an efficient FORCE learning algorithm (instead of Least Mean Squares – LMS – like in [13]), and does not produce semantic word labels as outputs: it is directly grounding a sentence into a semantic space. The learning is only done on the last time step, after having read the whole sentence.

E. Corpus generation (inputs and teacher signals)

To evaluate the model, a set of sentences associated to a simulated vision representation was created. Each evaluation was done by creating randomly a sentence with the grammar in Figure 3. Half of the time, a sentence was chosen uniformly among the sentences mentioning one object, for the other half, a two objects sentence was chosen. Once the sentence is chosen, a simulated vision precept corresponding to the sentence is generated, so that the sentence is a complete or partial description of the perceived scene. Objects described by the sentence are created, and if the sentence does not describe one aspect of the object, it is uniformly chosen among the possible values. In the case of sentences only mentioning one object, another object is created randomly one half of the time.

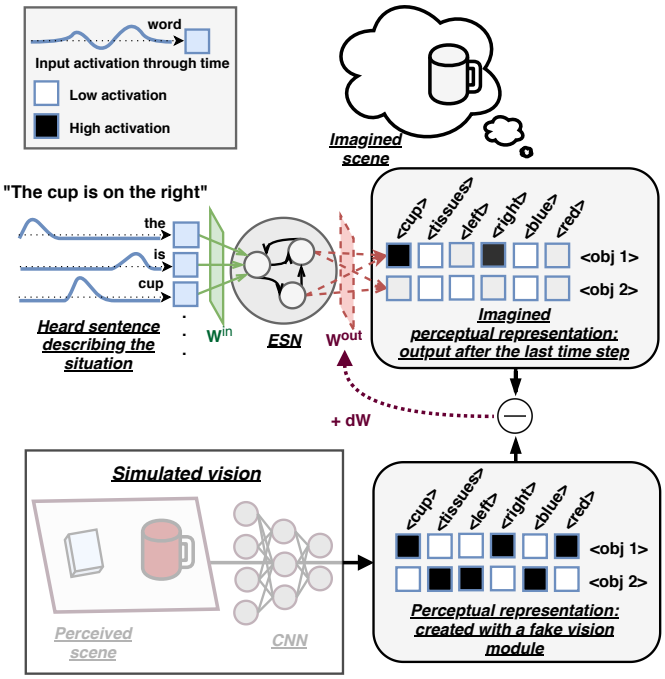

Fig. 2. ResPars adaptation to cross-situational learning. The network is fed a sentence describing a scene it perceives. The difference between the network output and the vision module output is used to update slightly the network weights. By repeating this step on various situations, only the pertinent modifications should be kept, and the network would extract the semantic of the sentences.

$$
\begin{aligned}
\text{OBJ} &\rightarrow \textbf{glass} \mid \textbf{cup} \mid \textbf{bowl} \mid \textbf{orange} \\
\text{COL} &\rightarrow \textbf{red} \mid \textbf{green} \mid \textbf{blue} \mid \textbf{orange} \\
\text{POS} &\rightarrow \textbf{left} \mid \textbf{right} \mid \textbf{middle} \mid \textbf{center} \\
\text{THE} &\rightarrow \textbf{a} \mid \textbf{the} \\
\text{THIS} &\rightarrow (\textbf{this} \mid \textbf{that}) \\
\text{SENTENCE-1-OBJ} &\rightarrow \text{THIS } \textbf{is} \text{ THE (COL)? OBJ} \\
&\quad \mid \text{THE OBJ (}\textbf{on the}\text{ POS)? } \textbf{is} \text{ COL} \\
&\quad \mid \text{THE (COL)? OBJ } \textbf{is on the} \text{ POS} \\
&\quad \mid \textbf{on the} \text{ POS (}\textbf{there}\text{)? } \textbf{is} \text{ THE (COL)? OBJ} \\
\text{SENTENCE} &\rightarrow \text{SENTENCE-1-OBJ} \\
&\quad \mid \text{SENTENCE-1-OBJ } \textbf{and} \text{ SENTENCE-1-OBJ}
\end{aligned}
$$

Fig. 3. Grammar used to generate the corpus. The grammar can generates sentences describing one or two objects. The total number of different sentences that could be generated is 775280 ($= 880 + 880^2$).

Thus, the created scenes contain one or two objects, and are associated with a sentence describing at least some aspects of the scene. The first mentioned object is always the first in the vector of the vision representation. We will discuss this choice in the discussion.

F. Input preprocessing before enterring the reservoir

In order to feed the network a sentence, a transformation of the sentence into a sequence of vectors is done. Sentences

are split into a list of words, with a *BEGIN* word and *END* word added. Each word is associated with a *one-hot* encoding vector, which means that network input is as big as the total number of encountered words: if a never met word appears, network input dimension is increased, and new random input weights are chosen for the word. Thus, a sentence is coded with the list of the one-encoded vectors corresponding to each word. Each one of these vectors is fed to the network during one step time, and only the final state and output layer are used for the network answering and learning.

G. Extraction of concepts from reservoir outputs

After having the network producing an output while hearing a sentence, the final step output activity is transformed into an object describing the sentence content. Network output dimension constraints the maximum number of objects that can be described in a scene. Each described object has several characteristics (category, position and color) which can be chosen among different concepts (e.g. $< left >$, $< middle >$, $< right >$) . The concepts associated with the same characteristic are mutually exclusive: only one at most is chosen to create the output scene.

To be selected in the created scene, a concept must have its associated network output activation satisfying three criteria. It must (i) be the highest among concept outputs from the same characteristics, (ii) be higher than a threshold T_o, and also (iii) be greater than a threshold T_s after the vector of the outputs from the same characteristics is passed to a *softmax* function, parameterized with an inverse temperature β. This allows to make sure that a selected concept is active enough, and that every other concepts from the same characteristic have a much more lower activity.

H. Evaluation and experiments

We performed a general experiment to validate that the model was able to learn with cross-situational learning. The performance evaluation is done on sentences not seen during learning, and parameters used for experiments are provided in Table I. Since the dataset is too huge, only a number of 1000 test sentences are selected each time (with the same random process as during learning) to evaluate the performance. Performance evaluation is averaged with 20 different instances of ESN, learning and being tested with different sentence sets, randomly chosen each time.

Two metrics, illustrated by Figure 4, are used to evaluate a network. Network imagined vision is considered *valid* if every concept mentioned in the sentence is present, and *exact* if they are also the only one present in the imagined scene. The percentage of sentences from the testing set considered as not *valid* or not *exact* is then used as quantitative error measurement.

The online FORCE learning method was compared with the classic ridge regression offline learning, using the same hyper-parameters in both situations. The network results were also compared to the expected error of a simple theoretical model which processes sentences by extracting content words (name, color, position) and their locations in the sentence structure. For instance, the sentence *"The cup is red"* would be transformed into *"x cup x red"*. A sentence describing two objects is split into two different single-object sentences that are processed separately. During the learning phase, the model only stores the result of the sentence processing. During the testing phase, it only returns a correct answer if the sentence matches one encountered during training. For instance, the sentence *"The cup is red and a bowl is on the left"* would be correctly treated if the model had already stored both sentences *"x cup x red"* and *"x bowl x x x left"*. The equation 6 computes the theoretical error of such a model trained on n_{train} sentences on the grammar from Figure 3 which generates n_{obj} different object name categories.

$$err_{\text{th}}(n_{\text{obj}}, n_{\text{train}}) = \left(1 - \frac{1}{85 \times n_{\text{obj}}}\right)^{2 \times n_{\text{train}}} \quad (6)$$

$85 \times n_{\text{obj}}$ comes from the general formula $[(n_{\text{col}} + 1) \times n_{\text{obj}}] + [n_{\text{obj}} \times (1 + n_{\text{pos}}) \times n_{\text{col}}] + [(n_{\text{col}} + 1) \times n_{\text{obj}} \times n_{\text{pos}}] + [n_{\text{pos}} \times 2 \times (n_{\text{col}} + 1) \times n_{\text{obj}}]$, which enumerates the different possibilities of simple sentences that the grammar from Figure 3 provides, with n_{col} and n_{pos} both set to 4. In Subsection III-D we give a qualitative analysis of the behavior of the model.

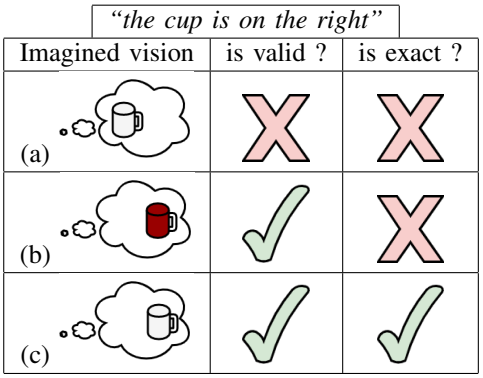

Fig. 4. Evaluations of different imagined scenes. (a) is not valid or exact because the cup is not on the right. (b) is not exact because the sentence does not mention the cup color.

Hereafter, we describe the experiments performed.

Experiment I: Number of reservoir units. The first experiment studies the influence of the number of neurons in the recurrent layer (i.e. the reservoir size) on the performance after a training session of 1000 sentences. Recurrent layers of sizes going from 10 to 1000 were tested. Results are provided in Subsection III-A.

Experiment II: Number of training sentences. In a second experiment, we studied the influence of the training set size on the performance, which was evaluated from 0 to 1200 training sentences every 50 sentences, with a reservoir with a 1000 neurons in its recurrent layer. The performance is compared to the abstract model. Results are provided in Subsection III-B.

Experiment III: Number of objects/concepts. The last experiment was about the model capacity to deal with a lot

of concepts. The number of different object categories and name was tested from 4 to 50, after having trained a reservoir with 1000 recurrent neurons on 1000 sentences. Results are provided in Subsection III-C.

I. Parameters

Hyperopt Bayesian search was used to find the optimum hyper-parameter set for the network. The library was given a cost function and a set of parameters to explore, described by Figure I, in order to find which ones were minimising the function. The library also allowed to visualize the effect of hyper-parameters on cost function, as shown in Figure 5. The cost function was instantiating a random reservoir with a recurrent layer size of 300 and an input matrix scaling of 1, and was evaluating the proportion of exact sentences on a testing set of size 200 after a training session of 500 sentences. The small numbers involved allowed to proceed 1000 evaluations of the cost function.

Hyper parameter	**Searching range**	**Optimium value**
Spectral radius ρ	$[0.3; 3]$	1.1
Leaky integrator α	$[0; 1]$	0.05
Recurrent layer density d	$[0; 1]$	0.85
Regularisation term ε	$[10^{-4}; 0.8]$	3.16×10^{-3}
Output activation threshold T_o	$[0; 1]$	0.6
Softmax activation threshold T_s	$[0; 1]$	0.1
Softmax inverse temperature β	$[0.5; 5]$	2.2

TABLE I
RANGE AND OPTIMUM VALUES OF HYPER-PARAMETERS VALUES USED IN THE HYPEROPT'S BAYESIAN SEARCH FOR THE MODEL SELECTION.

Optimum parameters found in table I were used with a reservoir of size of 1000 for the different experiments (if not stated otherwise). A graphical view of the hyper-parameter search can be seen in Figure 5.

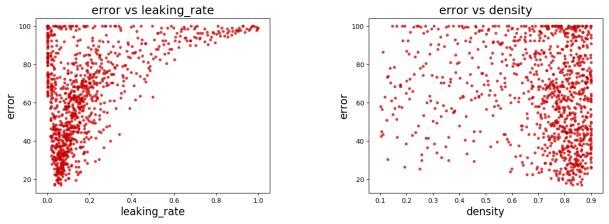

Fig. 5. Hyper-parameter vs. error after 1000 evaluations of different networks. For two hyper-parameters that we explored, the error for all simulations is plotted. Each dot corresponds to the error for one reservoir instance.

III. RESULTS

A. Experiment I: Number of reservoir units

Figure 6 shows the influence of the reservoir size (i.e. number of units in the reservoir) on the test error. Above 500 recurrent units, both error curves stabilize. One can see that errors from ridge regression learning method and FORCE learning method are the same.

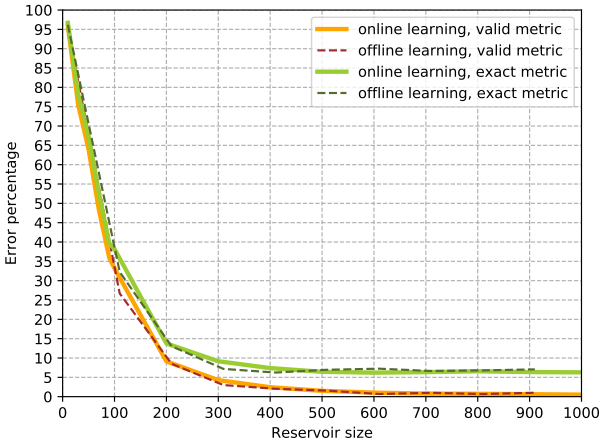

Fig. 6. Test error as a function of the number of units in the reservoir. Number of reservoir units: 1000. Number of objects/concepts: 4.

B. Experiment II: Number of training sentences

Figure 7 shows the influence of the number of sentences in the training set on the performance on the testing set. After 500 sentences presented, the reservoir shows 5% of error for *valid* evaluation and 14.2% of error for *exact* evaluation. After 1000 sentences presented, the reservoir shows 0.6% of error for *valid* evaluation and 6.3% of error for *exact* evaluation. The figure shows the equivalence between offline ridge regression and online FORCE learning method. Furthermore, one can notice the perfect matching of the error curves for the theoretical model error and the valid representation metric. This suggests the model is able to extract and remember the relevant information contained in the sentences it encountered in order to give back what was its perception at this moment.

C. Experiment III: Increasing of the number of objects/concepts

Figure 8 shows the influence of the number of possible word objects in input sentences (respectively the number of concepts in output) on the testing set error. The error is again exactly identical between offline ridge regression and online FORCE learning. Error curves seem to be composed of two "trends": at about 33 objects are presented, the increase of the error accelerates. This could be due to the model reaching a memory representation limit (due to the limited number of recurrent units): at such point, the latent space dimension of the reservoir seems too small to accurately represent/discriminate different inputs to perform the task. In order to continue on the first trend, one would need more recurrent units or extended memory components such as *Working Memory units* [24].

One can notice the different behaviours of the theoretical model error and the valid representation metrics. While the error is the same for a small number of objects (which corresponds to the curves match in Figure 8), the real model's performance gets significantly better when the number of

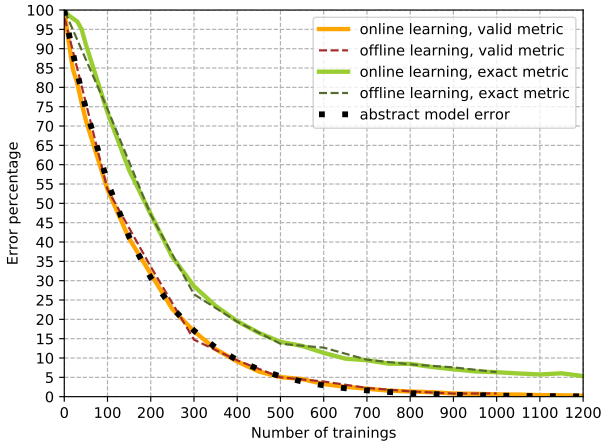

Fig. 7. Test error as a function of the number of sentences used for training. Performance of online and offline learning are equivalent for both metrics. A theoretical error function, representing an abstract simple model, is also plotted: one can notice it matches perfectly the curve of the more compliant metric. Error values are averaged over 20 simulations (with different random instances of reservoirs). Number of reservoir units: 1000. Number of objects/concepts: 4. Number of sentences in test set: 1000.

objects is contained between 15 and 40. This suggests the model have generalization abilities which allow it to performs better than the theoretical one. Because the simple theoretical model cannot generalise on unseen sentences, it is unable to perform well when a lot of combinations are possible.

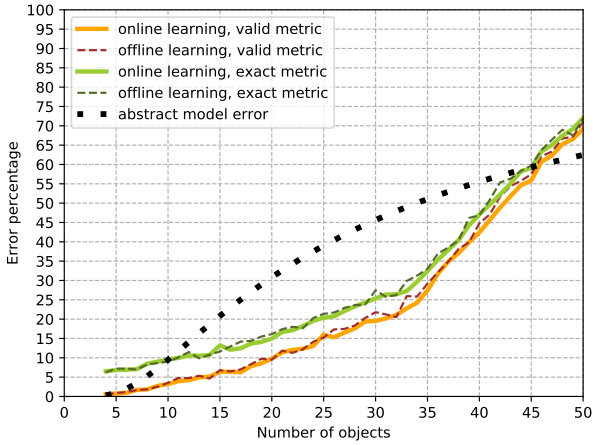

Fig. 8. Test error as a function of the number of possible word objects in sentences (i.e. possible object concepts in output). Performance of online and offline learning are equivalent for both metrics. A theoretical error function, representing an abstract simple model, is plotted: the real model performs significantly better than the theoretical one for a number of object contained between 15 and 40. Number of reservoir units: 1000. Number of sentence in training set: 1000.

D. Qualitative analysis

Figure 9 shows the reservoir outputs for a complex sentence with two objects. All output concepts corresponds exactly to what is expressed in the sentence, therefore this correspond to an *exact* output. In such case, *valid* and *exact* errors are both equal to zero. Moreover, the figure shows the reservoir outputs for a complex sentence with two objects which includes two times the same word *orange*, once as a noun and once as an adjective. This demonstrates that the reservoir learned the complex relationships between the contexts in which a word appears and their corresponding meaning concepts. For all qualitative figures, we applied a sigmoid function on the outputs in order to bound the outputs between 0 and 1, and to keep curves easy to visualize and interpret.

E. Complementary experiment

One can find the predictions done by the network before the end of the sentence (e.g. see curves of Figure 9) not always insightful. This is due to the fact that the training (i.e. weight modification) is performed only at the last time step (i.e. when *[END]* is presented) of each sentence.

We obtained more meaningful curves with subsidiary experiments. The learning procedure is slightly changed. After having trained the network with one sentence, the network is then trained on each word (of the sentence) presented separately. In this way, the network also learns correlations between words and visual representations. As shown on Figure 10 are much more meaningful and interpretable. Interestingly, when training the network with both usual and single-word sentences, outputs of the network provide consistent predictions during the whole presentation of sentences. The final answer from the network can be predicted before the sentence is over given its on-going activations. The output activity of a concept is activated once a word is pronounced. Performance-wise, results are a bit less efficient and training phase is more time consuming.

These curves are similar to the optimum curves obtained with offline supervised learning and by training over the full sentence (i.e. continuous learning) [12], [25]. We already obtained similar curves with online learning and *end-of-sentence* training with LMS [13] but the prediction curves were less accurate than in Figure 10. Therefore, it is surprising to obtain better shaped curves with more noisy learning (i.e. cross-situational learning) as several teacher outputs concepts do not correspond to the available concepts in a given sentence.

IV. Discussion

A. General comments

We have presented a model of recurrent network that learns to map a complex sentence describing one or several objects (e.g. *"An orange cup is on the right and there is an orange on the left."*) into a visual concept representation which provides the features of these objects. Training of the network is performed using a simulated vision module and cross-situational learning: after a few learning steps, the co-occurrence between perceived characteristics and the sequence

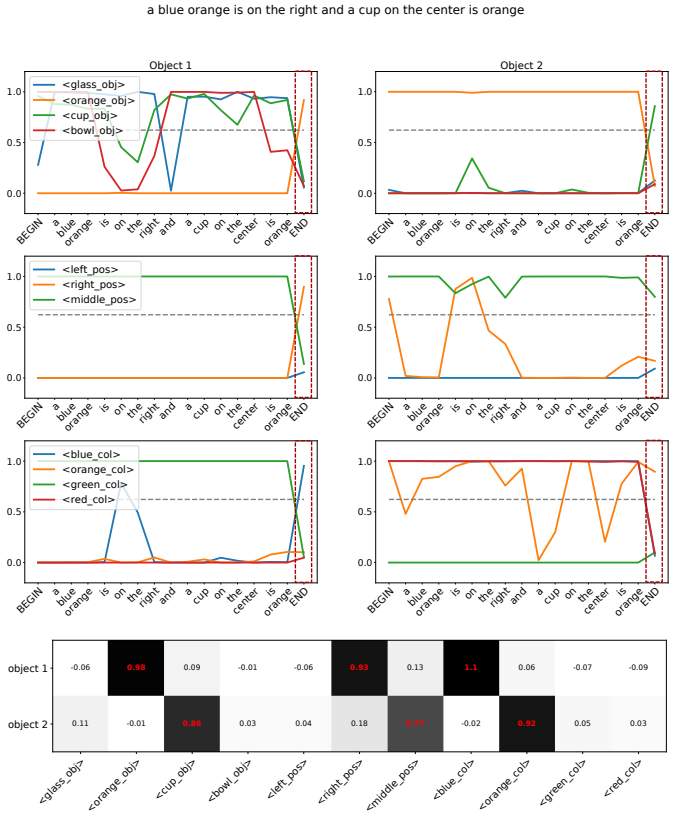

Fig. 9. The network is able to understand a sentence with two times the word orange, once as a noun and once as a adjective. This demonstrate that the reservoir learned the complex relations between the contexts were a word appears and their corresponding meaning concepts. This is an *exact* output, because all output concepts correspond exactly to what is expressed in the sentence. Actually, one can see that for most concept categories, the correct output is changing at the last time step (at the *END* marker input). This is a side effect of having the training only at the very end of the sentence. This randomly generated sentence reminds the one the French poet Paul Eluard "Earth is blue like an orange" [26].

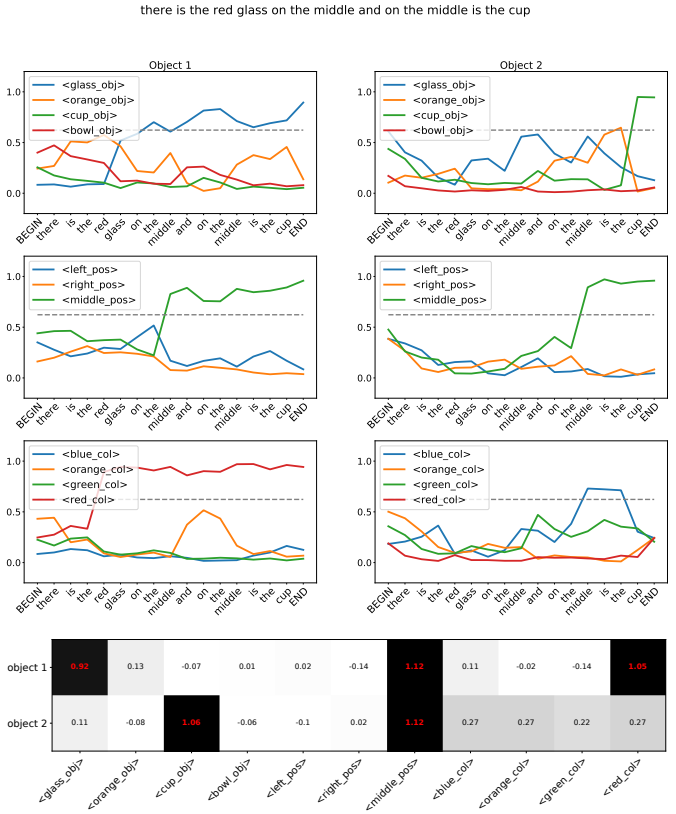

Fig. 10. Output curves obtained with a variant algorithm: not only the whole sentence but also each individual word are learnt in cross-situational learning. This results in an activity that "wakes up" once the word is fed to the network. Thus, the network provides a partial understanding during the unfolding of the sentence.

of words allows to identify which part of the sentences are associated with the visual information.

This study demonstrates that the proposed model, even if trained by cross-situational learning, is able to learn complex relations between the contexts in which a word appears and their corresponding meaning concepts (e.g. *orange* as an adjective or *orange* as a noun). In other words, it can learn that a subtitle change in the word order (e.g. *a blue orange* or *an orange cup*) can change the concept that should be activated. Surprisingly, it is able to perform this complex mapping only by being trained at the very last time step of the sentence presentation. Together with the fact that the model is able to learn in an online fashion and with few hundreds of sentences, it shows a new kind of capabilities of reservoirs that were not demonstrated previously to our knowledge.

In this paper, we investigated a case of noisy supervised learning with a cross-situational learning paradigm. However, children acquire language by making trials and guesses, e.g. when producing a sentence, or less perceptibly, when inferring what could be the meaning of an utterance. In future work, we will explore how reservoirs could be used to learn language in a reinforcement learning fashion.

B. Potential improvements of the architecture

More complex experiments could be done to evaluate limitations and more abilities of the model. Improvement of sentence complexities could be achieved, and biases between concepts be introduced in the training set (e.g. "cups are always orange"). Thus, we could check if the model is able to infer statistics about object features when they are not mentioned for instance.

Subsection II-F details how words are encoded, with *one-hot* vectors, obliging the model to remember every encountered word (in order to know which input index it corresponds to) and also to increase the network input dimension when new words are presented. This could be avoided by using a multidimensional hash function, allowing to map a word with a unitary vector of the same dimension of the input layer. Thus, in equation (1), $\mathbf{W}^{in}\mathbf{x}$, which is equivalent to $\mathbf{W}^{in}\mathbf{one\text{-}hot}(word)$ in our case, could be replaced by $\mathbf{W}^{in}\mathbf{hash}(word)$. The resulting implementation would be in theory equivalent.

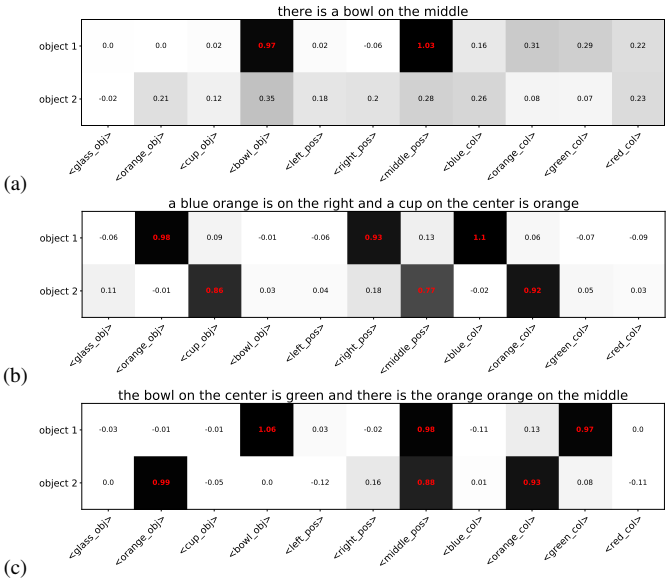

Fig. 11. (a) Simple sentence with *exact* comprehension (i.e. all activities reflect words of the sentence) (b) The word orange is used twice in the sentence: once as an adjective and once as a noun. Same sentence as Figure 9. (c) First, both words *center* and *middle* are recognized as the same concept $< middle >$. Secondly, the model is able to correctly understand the bi-gram *orange orange*: i.e. the repetition of the word *orange* is correctly interpreted.

Furthermore, since the output representation is independent from the sentence input representation, providing an utterance as a list of words is arbitrary. It could possibly be replace by providing an utterance as a list of phonemes (of phones), thus allowing to have a small input dimension (about 40 in English) and a more biologically plausible model.

In Subsection II-E we explained that when generating the training data, the first object mentioned in the input sentence is always in the first slot of the vision vector target. Thus the learning could not be purely unsupervised, since the order of the objects in the sentence must be known to create the output teacher. In order to remove this bias and have an unsupervised learning method, a different coding of the simulated vision scene should be found. A simple solution could consist in having a slot per position: but this would generate issues when sentences do not mention a position in the space (e.g. left/middle/right). Another solution could be the use of an auto-encoder to compress images into a latent representation, making the ESN directly outputing the scene features of the latent representation. Furthermore, the ESN outputs could be processed through the decoder part of the auto-encoder in order to obtain a picture of what is understood from the sentence. Similarly, Generative Adversarial Networks could be used in such perspective.

Once an unsupervised solution is found, the model could be plugged to a real vision module in order to test it. Morevoer, by including the notion of action in the latent representation, the model could understand sentences containing action commands. Thus, such model could be implemented into a self-learning robot, which could infer which command should be performed by some mechanism based on cross-situational learning along with reinforcement learning.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[2] K. Cho et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proc. of EMNLP*, pp. 1724–1734, 2014.

[3] R. Jozefowicz et al. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pp. 2342–2350, 2015.

[4] I. Sutskever et al. Sequence to sequence learning with neural networks. In *Proc. of NIPS*, pp. 3104–3112, 2014.

[5] A. Karpathy et al. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.

[6] J. R. Saffran et al. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928, December 1996.

[7] J. M. Siskind. A computational study of cross-situational techniques for learning word-to-meaning mappings. *Cognition*, 61(1-2):39–91, October 1996.

[8] L. Smith and C. Yu. Infants rapidly learn word-referent mappings via cross-situational statistics. *Cognition*, 106(3):1558–1568, March 2008.

[9] C. Yu and D. H. Ballard. A unified model of early word learning: Integrating statistical and social cues. *Neurocomputing*, 70(13-15):2149–2165, August 2007.

[10] O. Roesler et al. A probabilistic framework for comparing syntactic and semantic grounding of synonyms through cross-situational learning. In *ICRA-2018 Workshop on "Representing a Complex World: ..."*, 2018.

[11] A. Taniguchi et al. Cross-situational learning with bayesian generative models for multimodal category and word learning in robots. *Frontiers in Neurorobotics*, 11, December 2017.

[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*, 8(2):e52946, 2013.

[13] X. Hinaut and S. Wermter. An incremental approach to language acquisition: Thematic role assignment with echo state networks. In *International Conference on Artificial Neural Networks*. Springer, 2014.

[14] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. Technical Report 148, German National Research Center for Information Technology GMD, Bonn, Germany, 1 2001.

[15] M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[16] D. Sussillo and L. F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, Aug 2009.

[17] P. Dominey et al. A neurolinguistic model of grammatical construction processing. *Journal of Cognitive Neuroscience*, 18(12):2088–2107, 2006.

[18] X. Hinaut et al. Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Frontiers in Neurorobotics*, 8, 2014.

[19] J. Twiefel et al. Using Natural Language Feedback in a Neuro-inspired Integrated Multimodal Robotic Architecture. In *Proc. of RO-MAN*, New York City, USA, 2016.

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

[21] X. Hinaut and J. Twiefel. Teach your robot your language! trainable neural parser for modelling human sentence processing: Examples for 15 languages. *IEEE Transactions on Cognitive and Developmental Systems*.

[22] X. Hinaut and M. Spranger. Learning to parse grounded language using reservoir computing. In *Proc. of ICDL-Epirob*. IEEE, August 2019.

[23] M. Spranger et al. Open-ended procedural semantics. In *Language grounding in robots*, pp. 153–172. Springer, 2012.

[24] A. Strock et al. A simple reservoir model of working memory with real values. In *Proceedings of IJCNN 2018 - International Joint Conference on Neural Networks*, 2018.

[25] X. Hinaut et al. 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.

[26] P. Éluard. L'amour de la poésie. volume VII. 1929.