ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Semantic Role Labelling for Robot Instructions using Echo State Networks

Johannes Twiefel and Xavier Hinaut and Stefan Wermter *

Universität Hamburg - Department Informatik
Vogt-Kölln-Straße 30, D-22527 Hamburg - Germany
https://www.informatik.uni-hamburg.de/wtm/

Abstract. To control a robot in a real-world robot scenario, a real-time parser is needed to create semantic representations from natural language which can be interpreted. The parser should be able to create the hierarchical tree-like representations without consulting external systems to show its learning capabilities. We propose an efficient Echo State Network-based parser for robotic commands and only relies on the training data. The system generates a single semantic tree structure in real-time which can be executed by a robot arm manipulating objects. Four of six other approaches, which in most cases generate multiple trees and select one of them as the solution, were outperformed with 64.2% tree accuracy on difficult unseen natural language (74.1% under best conditions) on the same dataset.

1 Background and Related Work

1.1 Method: Echo State Networks

ESNs belong to the domain of reservoir computing and were e.g. described by Jaeger [1]. The main part of an ESN is called a reservoir, which is a recurrent neural network consisting of randomly and sparsely or (in our case) fully connected neurons. A special property of ESNs compared to other recurrent neural networks is the learning principle which leaves the input layer and the reservoir untrained. Only the output layer (*readout*), which is connected to the reservoir, is trained on reservoir states usually by *Linear* or *Ridge Regression* but it can also be trained online using *Least Mean Square* [2]. To be able to balance the influence of new inputs and past states of the network, we used leaky integrator neurons in the reservoir update equation [3]:

$$x(n+1) = (1-\alpha)x(n) + \alpha f(Wx(n) + W^{in}u(n+1) \tag{1}$$

with $x(n)$ the current state; W the weights inside the reservoir; W^{in} the weights of the input layer; $u(n+1)$ the next input; f the activation function ($tanh$); α the leak rate. The outputs of an ESN are given by $y(n)$, the output weights W^{out} are calculated using *Linear Regression*

$$y(n) = W^{out}z(n); W^{out} = (X^TX)^*X^TY \tag{2}$$

*This research was partially supported by a Marie Curie Intra European Fellowship within the 7th European Community Framework Programme: EchoRob project (PIEF-GA-2013-627156).

with W^{out} the weights of the output layer; $z(n)$ given by $[x(n); 1]$; X the states of the reservoir concatenated with a bias $([x(n); 1])$; Y the desired output of the system; $(X^T X)^*$ the *Moore-Penrose-Pseudoinverse* of $(X^T X)$.

1.2 Task: Supervised Semantic Parsing of Robotic Spatial Commands

The *SemEval-2014 Task 6* [4] was a contest in 2014 to provide parsing systems for a robotic spatial command dataset to extract semantic information in predefined structures. It was based on the *Train Robots* [5] dataset which consists of robot instructions and their corresponding semantic representation. Instructions were produced by internet users and are often grammatically incorrect. Whereas the whole treebank consists of around 10,000 sentences, for *SemEval-2014 Task 6*, a subset of 3,409 sentences from the treebank was chosen taking the first 2,500 sentences as training data, and the remaining 909 sentences as testing data. The dataset contains linguistically rich sentences, including ellipses, anaphoric references, multi-word spatial expressions and lexical disambiguation. The commands are related to a simulated environment containing an 8 x 8 board, on which differently colored objects like boxes and pyramids are placed (see Fig. 1). A robot arm is able to grasp objects from the board and move them to different positions. The sentences of the dataset contain a visual description of the scene before (left) and after (right) performing the desired action. For each sentence a semantic annotation exists which is described by the Robot Control Language (RCL). One challenging example sentence would be *"Pick up the blue block from the top of the green block and put it down on the blue block which lies next to another green block."*

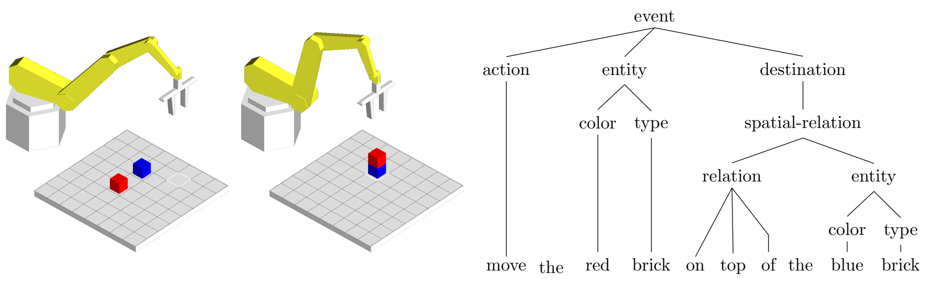

Fig. 1: An example for a board scene before (left board) and after (right board) the command *Move the red brick on top of the blue brick* and the corresponding parse tree.

1.3 Related Reference Systems

UW-MRS: Packard [6] developed a parser that uses the English Resource Grammar (ERG) as first-phase processing and employs a modified Berkeley

Parser as a second-phase backup. The ERG produces Minimal Recursion Semantics (MRS) as output and converts these to RCL statements. If the produced RCL statement is invalid, the Berkeley Parser creates hypothetical trees.
RoBox: The parser introduced by Evang and Bos [7] employs Combinatory Categorial Grammar (CCG) as a representation instead of RCL trees for training. To choose the best hypothetic tree, a structured perceptron is performing a post-processing step.
Tag&Parse: The *Tag&Parse* approach [8] consists of an independent tagger creating part-of-speech tags and performs chunking and chunk labelling. Then a constituency parser is used to generate RCL trees. To resolve anaphors, a maximum entropy model is used to generate tree hypotheses. Finally, the most probable tree is selected by the system.
KUL-Eval: Mattelaer et al. [9] developed a system using RCL statements as λ-expressions together with a probabilistic CCG.
Shrdlite: Shrdlite, a model of Ljunglöf [10] consists of a hand-written ambiguous grammar to generate multiple trees. The system selects the tree containing the minimum number of nodes as the best hypothetic tree.
UWM: The system developed by Kate [11] consists of the KRISP parser which uses a Support Vector Machine and a handwritten context-free grammar.

2 Approach

The aim of the developed approach is to provide a neural model for processing commands directed to a robot and create semantic representations that can be interpreted by that robot to perform the action contained in a given command. The developed model is inspired by the model of Hinaut and Dominey [12, 13], which is able to learn predicates from sentence structures consisting of closed class word (like *the, and, ...*) and place-holder open-class words. As input, our model receives tagged and labelled chunks (TLCs) (e.g. *action: take*). The TLCs are fed as a sequence to the network. At each timestep, only two neurons are active, one specifying the current tag (e.g. *action*), the other one setting the current label (e.g. *take*). For training, the reservoir state at the end of the sequence is collected for each input sequence and a *Linear Regression* is used to calculate the weights for the *readout*. The output nodes are used to generate RCL trees. Each word can have multiple outputs active. The possible outputs are *spatial-relation* (4), *sequence* (1), *destination* (1), *entity* (6), *measure* (1), *event* (2), *type-reference* (2*6) which are in total 27 for each word and at maximum 1080 for the longest assumable sentence (40 words). An output will be considered as active if its response is greater than 0.5. There are multiple outputs for several RCL elements to be able to distinguish elements of the same type. The inputs are generated by a second order Hidden Markov Model (HMM) [14], which takes words as input and generates *Inside Outside Beginning 2* (IOB2) representations [15]. The same model is used to generate chunk labels from words. Fig. 2 shows an example of processing a very simple sentence.

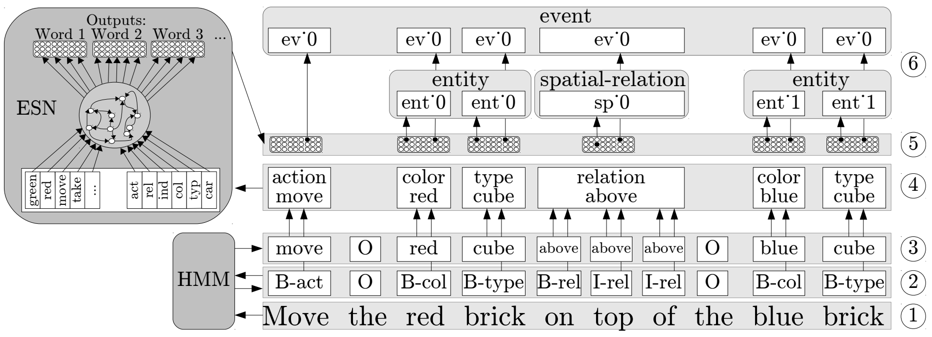
Outputs:
Word 1 Word 2 Word 3 ...
ESN
green red move take ... act rel ind col typ car
HMM
event
ev˙0
entity
ent˙0
spatial-relation
sp˙0
ent˙1
action move
color red
type cube
relation above
color blue
move O red cube above above above O blue cube
B-act O B-col B-type B-rel I-rel I-rel O B-col B-type
Move the red brick on top of the blue brick
1 2 3 4 5 6

Fig. 2: Processing of a simple example sentence. First, IOB2 tags ② are generated (e.g. *move* → *B-action*) from the input sequence ①. Then the HMM creates chunk labels ③ for words ① (e.g. *on* → *above*). Afterwards the identified chunks ② are merged (e.g. *B-rel I-rel I-rel* → *relation*) and are paired with the corresponding chunk tags ③ to a TLC sequence ④. The TLC sequence is fed to our ESN-based model which generates outputs for each words ⑤. The active outputs ⑥ are merged (e.g. *ent˙0* ... *ent˙0* → *entity*) and ordered in a hierarchy derived from the training data (e.g. *event* over *entity*). From this hierarchy, a tree representation can be derived (*event* as root, *entity* as child of *event*...; see Fig. 1).

3 Experiments and Results

The training dataset [5] consists of 2,500 sentences and 2,500 RCL trees representing their semantic content, while the test set contains 909 sentences and 909 RCL representations. To measure the performance of our system, we consider the metric defined in *SemEval-2014 Task 6* [4], which takes a string representation of the generated RCL tree and compares it directly to the reference tree. This way, even trees containing small errors are considered as classified wrongly which makes it a hard metric. In *SemEval-2014 Task 6*, the performance of the different systems (see Sec. 1.3) is measured for performing the task with and without the assistance of a planner, which is able to validate if a tree is correct for a given scene. As our system works without consulting a planner, we compare to the performance of the other methods measured without the planner. The results are shown in Table 1. To find out the best hyper-parameters for our system, we performed a 10-fold cross-validation on the training dataset exploring the parameters *leak rate* (from 0.1 to 0.9 with step size 0.1), *spectral radius* (from 0.25 to 10 with step size 0.25) for 10 different reservoir instance with a size of 500. The grid-search led to the best choice of hyper-parameters (*leak rate* = 0.3, *spectral radius* = 6) and was performed relatively fast (18 hours) using GPU processing to parallelize the computations. We fixed the reservoir size to 500 to increase the computation time, as Lukoševičius et al. [16] claimed that hyper-parameters scale for different reservoir sizes on the same task. To show the behaviour of the system with a reservoir network size, we measured the test error for different reservoir sizes up to 20,000 neurons (see Fig. 3). To train the 2,500 sentences, the system just needs several seconds for reservoir sizes up to 5,000 neurons, for 20,000 it already takes 5 minutes, while testing a single sentence is possible in real-time even with a reservoir of 20,000 neurons.

Approach	Accuracy
UW-MRS	(90.50)
RoBox	79.21
Tag& Parse	60.84
KUL-Eval	57.76
Shrdlite	51.50
UWM	45.98
Our Approach	64.2
Our Approach (perfect TLCs)	74.1

Table 1: Performance on *SemEval-2014 Task 6* [4] data.

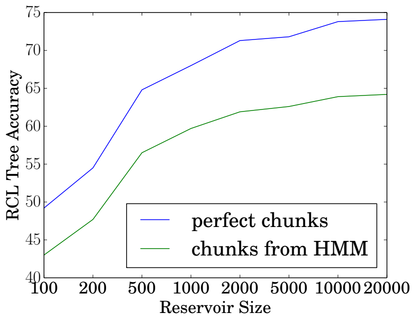

Fig. 3: Performance for different reservoir sizes.

4 Discussion

Compared to the other systems (see Sec. 1.3), which usually generate multiple possible trees and select one using a post-processor, our system only generates one tree as output, which leads to a constant computation time ($O(1)$) for a sentence with a given size, compared to $O(n^3)$ [17] for the CKY-based RoBox parser. Our ESN-based model is trainable in an efficient and fast way, as only the output weights have to be calculated only at the last time-step. Even scaling up the reservoir size up to 20,000 neurons (which leads to an increased performance (see Fig. 3)) does not prevent the system to work in real-time. A larger reservoir would be possible but is computationally too expensive, as with modern GPUs (approx. 5,000 cores) not all computations can be parallelized and a lot of data shifting has to be performed, which slows down the training process. The results in Table 1 show that the system outperforms four of the six reference systems. The best system (UW-MRS) is based on the English Resource Grammar which is an external expert, while our system only uses the training data as input. The system would be even better if the TLCs generated by the HMM were correct (64.2% $\rightarrow$ 74.1% accuracy). Because of this fact, we plan to develop an improved chunker based on ESNs. After inspecting the neural outputs of the system, we identified that often only one or few labels are incorrect, and that there are low activities for different concurrent outputs (e.g. *spatial-relation* and *entity*), which leads to the hypothesis that the system was trained with ambiguous data and provides weak responses to the different possible outputs at the same time. We plan to introduce a post-processing step which disambiguates these outputs. By solving this systematic error, the system could outperform the other methods which do not use an external expert. Due to the fact that the system is running in real-time, we are also planning to bring the system to a real-world scenario by connecting it to an existing speech recognition environment [18] and to employ the outputs of the system to control a real-world robot like in [13].

References

[1] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148:34, 2001.

[2] X. Hinaut and S. Wermter. An incremental approach to language acquisition: Thematic role assignment with echo state networks. In *ICANN 2014*, pages 33–40. Springer, 2014.

[3] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–352, 2007.

[4] K. Dukes. Semeval-2014 task 6: Supervised semantic parsing of robotic spatial commands. *SemEval 2014*, page 45, 2014.

[5] K. Dukes. Train robots: A dataset for natural language human-robot spatial interaction through verbal commands. In *ICSR. Embodied Communication of Goals and Intentions Workshop, Bristol, United Kingdom*, 2013.

[6] W. Packard. UW-MRS: leveraging a deep grammar for robotic spatial commands. *SemEval 2014*, page 812, 2014.

[7] K. Evang and J. Bos. Robox: CCG with structured perceptron for supervised semantic parsing of robotic spatial commands. *SemEval 2014*, page 482, 2014.

[8] S. Stoyanchev, H. Jung, J. Chen, and S. Bangalore. AT&T: The tag&parse approach to semantic parsing of robot spatial commands. *SemEval 2014*, page 109, 2014.

[9] W. Mattelaer, M. Verbeke, and D. Nitti. Kul-eval: A combinatory categorial grammar approach for improving semantic parsing of robot commands using spatial context. *SemEval 2014*, page 385, 2014.

[10] P. Ljunglöf. Shrdlite: Semantic parsing using a handmade grammar. *SemEval 2014*, page 556, 2014.

[11] R. J. Kate. UWM: Applying an existing trainable semantic parser to parse robotic spatial commands. *SemEval 2014*, page 823, 2014.

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

[14] K. Dukes. Contextual Semantic Parsing using Crowdsourced Spatial Descriptions. *arXiv:1405.0145 [cs]*, May 2014. arXiv: 1405.0145.

[15] E. F. Tjong Kim Sang and S. Buchholz. Introduction to the conll-2000 shared task: Chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th CoNLL-Volume 7*, pages 127–132. Association for Computational Linguistics, 2000.

[16] M. Lukoševičius. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade*, pages 659–686. Springer, 2012.

[17] X. Song, S. Ding, and C. Lin. Better binarization for the cky parsing. In *EMNLP 2008. Hawaii, USA*, pages 167–176. Association for Computational Linguistics, 2008.

[18] 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*, pages 1529–1535, 2014.