ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

An Analysis of Subtask-Dependency in Robot Command Interpretation with Dilated CNNs

Manfred Eppe, Tayfun Alpay, Fares Abawi and Stefan Wermter *

University of Hamburg - Department of Informatics
Vogt-Koelln-Str. 30 - 22527 Hamburg - Germany
http://www.informatik.uni-hamburg.de/WTM

Abstract. In this paper, we tackle sequence-to-tree transduction for language processing with neural networks implementing several subtasks, namely tokenization, semantic annotation, and tree generation. Our research question is how the individual subtasks influence the overall end-to-end learning performance in case of a convolutional network with dilated perceptive fields. We investigate a benchmark problem for robot command interpretation and conclude that dilation has a strong positive effect for performing character-level transduction and for generating parsing trees.

1 Introduction

Artificial neural networks frequently serve as sequence-to-sequence (seq2seq) transducers for language translation, parsing, and other natural language processing (NLP) tasks [7]. Despite their success, little is understood about the inner knowledge abstraction mechanisms of neural networks and the reason for their success, especially in end-to-end settings. These knowledge abstraction mechanisms implicitly involve several subtasks that traditional grammar-based approaches for parsing perform explicitly as pre- and postprocessing steps, namely tokenization, semantic annotation, and tree generation. However, current approaches lack a systematic study of how well each subtask contributes to the overall the performance. Such studies are necessary to understand which pre- and postprocessing methods are useful and to gain insight into the implicit symbol processing and abstraction mechanisms that take place in neural networks. Our work addresses this question with a systematic investigation about how the implicit learning of the individual subtasks affects the overall learning performance for a recently proposed network architecture, namely a Dilated Causal Convolutional Neural Network (DCCNN; [4, 6]).

Its inventors claim that the dilated perceptive fields allow the network to learn long-distance dependencies, which is important for NLP tasks like machine translation and generating parsing trees. The authors support their claim by evaluating standard benchmark problems like sentiment analysis. However, their evaluation does not analyze which of the individual parsing subtasks from the overall learning procedure are particularly hard or easy for the network to learn, and how the dilation influences the subtask-dependent performance.

*The authors gratefully acknowledge partial support from the German Research Foundation DFG under project CML (TRR 169), the European Union under project SECURE (No 642667), and the Hamburg Landesforschungsförderungsprojekt CROSS.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

The contribution of our paper is an analysis that investigates the influence of dilation in the context of different pre- and postprocessing subtasks. Specifically, we hypothesize that i) dilation has a positive effect on the transduction of long character-level sequences where tokenization is not performed as preprocessing step, and that ii) dilation is useful if trees are generated in an end-to-end manner, and not during an explicit postprocessing step.

In order to be able to draw meaningful conclusions, we concentrate on a small controllable dataset for robot command interpretation, namely the TrainRobots benchmark [2], where natural language commands must be parsed into a tree-structured well-defined Robot Control Language (RCL). We also demonstrate that our system performs well for this benchmark, but our main contribution is the subtask-dependent analysis of the learning process.

2 Dilated CNNs and robot command interpretation

Recent work on sequence transduction with neural networks (e.g. [8]) demonstrates its applicability to parsing, even though parsing implies the generation of tree structures. Sequential data can be used to represent trees in the form of a sequence of parse transitions [1] or as a string with dedicated delimiter symbols, e.g. parenthesis, to indicate parent-child relationships. In the case of parsing, we refer to the *source* as a natural language sentence and to the *target* as a parse tree. The training goal is to minimize the error between the target predicted by the neural network and the target of the training corpus. For the present study, we employ a DCCNN, a Convolutional Neural Network with causal dilations [4, 6]. Dilation is claimed to be the main reason for the DCCNN's capability to capture long-distance relationships since it increases the size of the perceptive field [4]. Conceptually, dilation means that only each d-th element of an input vector patch is part of a kernel, whereas in a standard convolutional layer every element is part of a kernel. By doubling the *dilation rate* d with each layer l, the perceptive field grows exponentially. To evaluate the effect of dilation in our experiments, we also investigate i) the case of a linearly increased $d = l$, which causes the dilated field to grow polynomially, and ii) the case where $d = 1$ is constant, i.e. no dilation is applied and the perceptive field grows linearly as in standard convolution. The architecture contains several blocks of dilated layers, followed by a final softmax layer to predict the token sequences. It uses the source and the prediction for a sequence prefix as input in order to generate the prediction for the next token and is therefore recurrent. In our work, we use the TrainRobots dataset to examine this method.

The TrainRobots dataset was introduced as Task 6 of the SemEval 2014 competition [2]. It consists of 3409 pairs of natural language robot commands and symbolic command instructions in RCL, with 2500 used for training and 909 for testing. RCL builds on a role-filler paradigm to assign tokens in the input sentence to leaves of a semantic tree. The trees are provided as strings, with parentheses as delimiters, indicating the start and end of nodes, as well as implicit edges. For example, consider the following string that represents the

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

parse tree of the sentence "place the green pyramid on top of [sic] red brick.":

```
(event: (action: move (token: 1)) (entity: (color: green (token: 3)) (type: prism
(token: 4))) (destination: (spatial-relation: (relation: above (token: 5 7)) (entity:
(color: red (token: 8)) (type: cube (token: 9))))))
```

The dataset contains crowdsourced realistic samples, often with ungrammatical and noisy language such as in the above example where the determiner "the" is omitted before the noun "red brick". Roles and fillers are assigned using position numbers. For example, `relation: above (token: 5 7))` denotes that the `above`-relation role is mapped to tokens five to seven of the input string.

The TrainRobots dataset is particularly useful for our purposes for three reasons. First, it is small and hence very controllable. This gives us stable conditions to easier investigate the different phenomena we aim to explore. Second, it involves the generation of tree-structured data with role-filler assignments, allowing us to draw conclusions that are concerned with how one could use a neural network that natively generates sequences to generate trees. Third, the dataset is concerned with a practical application, namely robot command interpretation, and there is a number of previous results produced by other approaches [3] for this dataset, enabling us to compare our work.

3 Subtasks of generating parsing trees

To generate the RCL trees, we identified and evaluated the following subtasks that the network implicitly learns as part of the overall parsing process:

Tokenization. Tokenization separates a string into meaningful chunks of symbols, i.e., words. With respect to neural networks, tokenization can be understood as a strong form of feature learning. We investigate how implicit tokenization influences the learning performance by performing manual tokenization as a preprocessing step and then performing sequence transduction on the token level. That is, we compare character-level transduction with word-level transduction and measure the improvement.

Semantic annotation. For successful parsing, the network implicitly learns to map words to specific semantic RCL types. For example, it learns that the words "prism" and "tetrahedron" both denote an object of the semantic type `pyramid`. To evaluate the effect of performing semantic annotation as preprocessing, we use the token assignment information from the target data and modify the source data so that all words for the same semantic type are replaced by the same word. For example, we replace all `pyramid` instances such as "prism", "pyramid", and "tetrahedron" with the same word "pyramid". This preprocessing step does not completely solve the semantic annotation problem but facilitates the task by learning one-to-one instead of many-to-one mappings.

Tree generation. As part of the parsing task, the network learns constraints in the tree structure. For example, it learns the constituency constraint that an `event` node always consists of three specifically ordered child nodes, namely firstly an `action` node, followed by an `entity` node, ending with a

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

`destination` node. Since these constraints are unambiguous for the Train-Robots dataset, the tree structure can actually be derived from the leaf nodes as their order of occurrence determines each parent node type. We, therefore, let the neural network generate a sequence of leaf nodes which we use to generate the parse trees during postprocessing.

Token assignment. Token assignment refers to mapping the tokens of an input string to the leaves in a parse tree. As seen in the above example from Sec. 2, this is realized by enumeration in RCL. For example, token 3 refers to the color green. Besides enumerations, we also investigate direct references as a method of token assignment. For these, we preprocess the RCL data and substitute the enumeration with the original tokens from the source sentence, e.g. `(relation: above (token: 5 7))` becomes `(relation: above (token: 'on top of'))`. We also investigate the command interpretation performance when no tokens are assigned.

4 Subtask-dependent empirical evaluation

To evaluate our approach, we compare how the different subtasks influence the learning performance for constant, linear, and exponential dilation rates (see Sec. 2). The results of all 72 experiments are depicted in Fig. 1. The evaluation metric is the average percentage of correctly generated parse trees. As hyper-parameters, we have empirically determined 8 dilated blocks of 4 convolutional layers for both encoder (kernel size 4) and decoder (kernel size 2) which are used consistently in all experiments.

We observe that all pre- and postprocessing measures yield significant improvements for the parsing. The best results (71.8%) are achieved when using semantic annotation and tokenization in a preprocessing step, and generating trees from the generated leaves during postprocessing. The tokenization effectively causes the network to perform word-level transduction. The comparison to character-level transduction, where not tokenization is performed, shows that dilation has a much stronger impact on in the character-level cases, which supports our first hypothesis. The experiments also support our second hypothesis and show that dilation has the largest positive impact in cases where generating the parse tree is done implicitly by the neural network in an end-to-end manner, and not explicitly during the postprocessing. This holds especially for the longer character-level sequences. Indeed, the impact of dilation on end-to-end cases for character-level transduction with implicit tree generation by the neural network is strong enough to achieve an accuracy of 71.1%, which is comparable to that of word-level transduction with tree generation as a postprocessing step.

A comparison of word-level vs. character-level transduction for cases where trees are not generated during postprocessing is also very interesting. This means that tokenization as preprocessing step effectively mitigates the accuracy instead of improving it. The transduction errors that we observed during a detailed investigation of such word-level cases were mostly related to confusing role instances. That is, the generated word level tree structure was mostly correct,

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

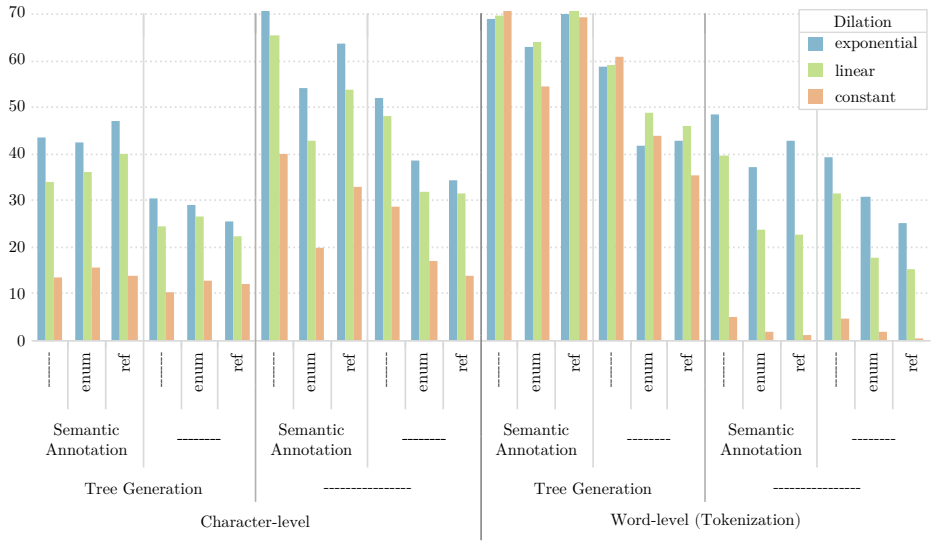

Fig. 1: Evaluation of learning to generate robot command language parsing trees with a DCCNN utilizing exponential, linear, and constant dilation rates. Bottom labels are the respective pre- or postprocessing steps as described in Sec. 3, i.e. from top to bottom: token assignment (by *enum*eration, *ref*erence, or not at all), semantic annotation (with and without), tree generation (with and without), and word-level tokenization (vs. character-level input). Displayed is the percentage of generated RCL commands that are fully correct.

but colors, objects and relations were often confused (e.g. "green" confused with "yellow" and "cube" with "pyramid").

We further observe that the difference between token assignment via reference compared to token assignment via enumeration is quite small. This demonstrates that neural networks are good at implicitly counting words, for both character- and word-level processing. As expected, when tokens are not assigned at all the best results are achieved. Semantic annotation, on the other hand, significantly improves the results of all experiments which is mostly unsurprising as it simplifies and reduces the number of mappings that the networks have to learn.

In addition to the investigation of the effect of pre- and postprocessing on language understanding and parsing, we have compared our approach to other participants of the TrainRobots benchmark. Our results are very competitive as our system does not use any hand-crafted background knowledge [3]. In fact, to the best of our knowledge, it is the currently best-performing system that does not use hand-crafted background knowledge. The currently best performing system achieves 90.5% accuracy, and it is a hybrid system that combines a rule-based parser based on a hand-crafted grammar with a learning approach. The second place achieves 79% accuracy by using only a hand-crafted grammar. The third best approach [9] does not use either but instead employs a combination of Hidden Markov Models and Echo State Networks, achieving 64.2% accuracy.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

5 Summary and Future Work

We have systematically evaluated several pre- and postprocessing methods that an external system can perform to improve the parsing with neural networks. The study concentrates on dilated causal convolutional networks, because they have proven to be very efficient, and, to the best of our knowledge, a systematic survey on how to preprocess data for efficient learning has not yet been published in this context. The central conclusion that we draw from our experiments is that dilated convolution is useful for processing and producing long sequences, such as in character-level and multi-sentence transduction. This is also important when using seq2seq transducers to generate trees that are encoded as relatively long sequences of delimiters and node labels.

Our investigations underpin the effectiveness of combining neural approaches for NLP with other symbolic or statistical methods, especially in cases where training data is only sparsely available. As an additional contribution, we present a system that, without extensive hyperparameter optimization, is competitive in the TrainRobots benchmark. As future work, we want to investigate further the parsing of larger paragraphs with several sentences which may include context. This is relevant for applications like situated robot command interpretation, where a command may depend on the context of earlier utterances, or on completely different modalities, like camera images or recognized gestures, that may also be part of the processed data.

References

[1] D. Chen and C. D. Manning, A fast and accurate dependency parser using neural networks. In Proc. *EMNLP*, pp. 740–750, 2014.

[2] K. Dukes, semantic annotation of robotic spatial commands. In Proc. *LTC*, pp. 319–323, 2013.

[3] K. Dukes, SemEval-2014 Task 6: Supervised semantic parsing of robotic spatial commands. In *SemEval@ COLING*, 2014.

[4] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. v. d. Oord, A. Graves and K. Kavukcuoglu, Neural machine translation in linear time. Technical Report, arXiv preprint arXiv:1610.10099, 2016.

[5] T. Mikolov, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality. In Proc. *NIPS*, pp. 3111–3119, 2013.

[6] A v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[7] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks. In Proc. *NIPS*, pp. 3104–3112, 2014.

[8] K. S. Tai, R. Socher and C. D. Manning, Improved semantic representations from tree-structured long short-term memory networks. In Proc. *ACL-IJCNLP*, pp. 1556–1566, 2015.

[9] J. Twiefel, X. Hinaut and S. Wermter, Semantic role labelling for robot instructions using echo state networks. In Proc. *ESANN*, pp. 695–700, 2016.