Convolutional neural networks for low-resource morpheme segmentation: baseline or state-of-the-art?

Alexey Sorokin,
Neural Networks and Deep Learning Laboratory, Moscow Institute of Physics and Technology
Faculty of Mathematics and Mechanics, Moscow State University
alexey at sorokin dot mail dot ru

Abstract

We apply convolutional neural networks to the task of shallow morpheme segmentation using low-resource datasets for 5 different languages. We show that both in fully supervised and semi-supervised settings our model beats previous state-of-the-art approaches. We argue that convolutional neural networks reflect local nature of morpheme segmentation better than other neural approaches.

Morpheme segmentation consists in dividing a given word to meaningful individual units, morphs, which are surface realizations of underlying abstract morphemes. For example, a word *unexpectedly* could be segmented as *un-expect-ed-ly*, and the morpheme *-ed* may be also realized as *-t* like in *learn-t*. The generated segmentation may be used as input representation for machine translation (Mager et al., 2018) or morphological tagging (Matteson et al., 2018) or for automatic annotation of digital linguistic resources. Briefly, information about internal morpheme structure makes the data less sparse since an out-of-vocabulary word may share its morphemes with other words already present in the training set. This helps to recover semantic and morphological properties of an unknown word, which otherwise will be unaccessible. The task of morpheme segmentation is especially important for agglutinative languages, such as Finnish or Turkish, where a word is formed by attaching a sequence of affixes to its stem. This affixes reflect both derivational and inflectional processes. A common example from Turkish is *evlerinizden* '*from your houses*', which is decomposed as:

ev	ler	iniz	den
house	+PL	*your*+PL	+ABL

The task of morpheme segmentation is even harder for polysynthetic languages: while in agglutinative languages morphemes are usually in one-to-one correspondence with morphological features, for polysynthetic languages this matching is more complex with no clear bound between compound words and sentences. For example, in Chuckchi language the whole phrase '*The house broke*' can be expressed as

ɣa	ra	semat	ɬen
+PF	house	break	+PF+3SG

Consequently, polysynthetic language demonstrate extremely high morpheme-to-word ratio, which leads to high type-token ratio, which makes their automatic processing harder. Even further, this processing is performed in low-resource setting since most polysynthetic languages have only few hundreds or thousands of speakers and consequently tend to lack annotated digital resources. Hence, the algorithms initially designed for less complex languages with more data (mostly for English) may change significantly their properties when applied to low-resource polysynthetic data. That is especially the case for neural methods, which are (often erroneously[1]) believed to be more data-hungry than earlier approaches.

However, in 2019 it is insufficient to just say "neural networks" in case of NLP, since there are various neural networks whose properties may differ significantly. Leaving aside the immense diversity of network architectures, they can be separated in three main categories: the convolutional ones (CNNs), where convolutional windows capture local regularities; the recurrent ones, where GRUs and LSTMs memorize potentially unbounded context; and sequence-to-sequence (seq2seq) models, which perform string transductions using encoder-decoder approach. Among the three, convolutional neural networks are the least

[1] see (Zeman et al., 2018) and (Cotterell et al., 2017) that show that both in morphological tagging and automatic word inflection neural networks are clearly superior, though their architecture should be adapted for the lack of data.

Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 154–159
Florence, Italy. August 2, 2019 ©2019 Association for Computational Linguistics

explored, however, we argue that they are more effective for surface morpheme segmentation.

In our work we support two claims: 1) convolutional networks improve seq2seq approaches for neural morpheme segmentation 2) language model trained on unlabeled data may be useful to further improve their performance. We apply our models to 4 indigenous languages, spoken in Mexico: Mexicanero, Nahuatl, Wixarika and Yorem Nokki, since the scores for them are available in recent studies Kann et al. (2018). We also test our approach on North Sámi data from Grönroos et al. (2019).

1 Related work.

Automatic morpheme segmentation was extensively studied in pre-neural years of modern NLP. The investigations had two principal directions: several researchers tried to implement the approach of Harris (1970) and Andreev (1965) to find a quantitative counterpart of morpheme boundaries in terms of letter statistics. These methods were mainly unsupervised and include the well-known Morfessor system: Creutz and Lagus (2002) and its successors Creutz and Lagus (2007) and (Virpioja et al., 2013) (the latter uses semi-supervised learning). There was also an extensive work in the field of adaptor grammars.[2](Johnson et al., 2007; Sirts and Goldwater, 2013; Eskander et al., 2018) However, both these approaches are generative by their nature and are based on a probabilistic model of word structure. The most successful pure machine learning method was CRF-based model designed in Ruokolainen et al. (2013, 2014), which still remains state-of-the-art on several morpheme segmentation datasets.

There were several attempts to apply neural networks for morpheme segmentation and closely related problem of word segmentation, which is enevitable for Chinese, Japanese and other languages with similar graphics. The first one was probably Wang et al. (2016), which used window LSTMs, latter works include Kann et al. (2016) and Ruzsics and Samardzic (2017) which applied the sequence-to-sequence approach. Our study is conducted on the material from Kann et al. (2018), where the sequence-to-sequence model with attention was applied to the material of 4 indigenous North-American languages, both is supervised and semi-supervised manner. All these studies solve morpheme segmentation as sequence transduction. In contrast, Shao (2017) treated morpheme and word segmentation as sequence labeling task which can be solved with BiRNN-CRF network.

p	r	e	t	r	a	i	n	s
B	M	E	B	M	M	M	E	S

Figure 1: Morpheme segmentation of word *pre-train-s* expressed with BMES scheme.

The main inspiration for our work is Sorokin and Kravtsova (2018), who demonstrated, that at least for Russian (a fusional language with lots of data available) convolutional neural networks significantly outperform all other approaches, also being the less data-consuming (see also (Bolshakova and Sapin, 2019) for detailed comparison). The recent study of Grönroos et al. (2019) modified the decoder in seq2seq architecture to make its independent of the previous timesteps, which makes their model essentially an LSTM-based sequence tagger.

2 Model architecture.

Basing on the ideas from Sorokin and Kravtsova (2018), we decide to refrain from seq2seq approaches and reduce the morpheme segmentation task to sequence labeling problem. We solve this problem using convolutional neural networks. Each segmentation in the training set is encoded using BMES-scheme as illustrated on Figure 1. Here, S denotes single-letter morpheme; in case the morph is at least two letters long B stands for morpheme beginning, E for its end and M for all interior letters. Thus, the task of the algorithm is to predict the sequence of labels given the sequence of letters (probably, enriched with special BEGIN and END symbols). Due to the local nature of CNNs, the model cannot see any symbols except those surrounding the current one. However, the width of this local window may be up to 9 letters,[3] which makes the model powerful enough to capture all relevant local context.

2.1 Basic model.

Our basic architecture closely follows the model of Sorokin and Kravtsova (2018). The input of

[2] Roughly speaking, an adaptor grammar tries to learn from data a probabilistic context-sensitive grammar for morph sequences.

[3] In case of two layers with convolution width 5.

the algorithm is a sequence of $0/1$-encodings, which are transformed to symbol embeddings by an embedding layer. These embeddings are passed through several stacked convolutional layers of different widths, as, for example, in Kim et al. (2016), the final outputs of all layers are concatenated. For better convergence we insert batch normalization and dropout layers between consecutive convoluions. The obtained context encodings are passed through a dense layer with softmax activation which generates a probability distribution over possible tags. Since not every sequence of tags corresponds to a valid morpheme segmentation, we find the most probable segmentation using Viterbi algorithm.

2.2 Multitask training and one-side convolutions.

Kann et al. (2018) demonstrates that pretraining on auxiliary task of autoencoding, which is the restoration of original input sequence, can be beneficial for morpheme segmentation. Autoencoding is an appealing complementary task since it does not require additional labeled data. It is especially suitable for encoder-decoder architecture since the memorization of input sequence is the natural job of the encoder. However, this objective does not fit in our paradigm since we try to avoid global architectures, such as recurrent ones and especially seq2seq, in favor of the local ones. Following modern trends in NLP, we select language modelling as an auxiliary task, predicting not only the morpheme boundary of the current symbol but also the following symbol. However, this approach fails with basic CNN architecture since the convolutional window observes the next symbol and can easily memorize it.

Therefore we slightly modify our model: instead of using a symmetric window around current symbol, we have two groups of convolutions: the left and right ones. The left observes the current symbols and also some symbols preceding it, while the right does not see preceding symbols, but only the current one and the ones following it. We again use windows of different size and concatenate their outputs, thus obtaining for each position t two context embeddings $\vec{h}_t$ (left) and $\overleftarrow{h}_t$ (right). They are used to obtain the required distribution $\mathbf{p}_t$ over morphological labels as well as two auxiliary distribution $\mathbf{q}_{t-1}$ and $\mathbf{q}_{t+1}$ over preceding and following symbols, respectively:

$$\begin{aligned}\mathbf{p}_t &= \mathrm{softmax}_{morph}(U[\vec{h}_t, \overleftarrow{h}_t]),\\ \mathbf{q}_{t-1} &= \mathrm{softmax}_{symb}(V_l \overleftarrow{h}_t),\\ \mathbf{q}_{t+1} &= \mathrm{softmax}_{symb}(V_r \vec{h}_t).\end{aligned}$$

Note that this architecture with “unidirectional” convolutions can be used without auxiliary objective as well.

3 Data.

We evaluate our model on two datasets: the dataset of 4 indigenous North American languages from Kann et al. (2018) and the North Sami dataset from Grönroos et al. (2019). In this section we briefly characterize the languages, for more complete description we refer the reader to the cited papers or to linguistic resources such as WALS(Haspelmath et al., 2005).

1. The 4 mexican languages: Mexicanero, Nahuatl, Wixarika and Yorem Nokki all belong to Yuta-Aztecan family. They are mostly agglutinative and have extremely complex verb morphology. Some stems and even affixes in case of Mexicanero are Spanish borrowings.

2. North Sámi is a Finno-Ugric language spoken in the North of Finland, Sweden, Norway and Russia. It is morphologically complex, featuring derivational, inflectional and compounding processes. It also has regular but complicated morphonological variation.

The quantitative characteristics of the datasets used in our study are given in Table 1. For mexican languages we used the same data as in Kann et al. (2018). The number of unlabeled words used for semi-supervised models differ because of different preprocessing.[4]

4 Experiments

4.1 Model parameters.

We use symbol embeddings of size 32. The basic model contains two parallel convolutional groups

[4] It is not an obstacle for fair comparison since the main goal of our paper is to compare supervised versions of the model.

[5] As in Kann et al. (2018), the same list of words is used for Mexicanero and Yorem Nokki due to their close relatedness.

[6] Actual word lists are larger but we restrict it to random 100000 words to speed up training.

Language	Train	Dev	Test	Unlabeled
Mexicanero	427	106	355	978[5]
Nahuatl	540	134	449	36149
Wixarika	665	176	553	13092
Yorem Nokki	511	127	425	978[5]
North Sámi	1044	200	796	100000[6]

Table 1: Size of the datasets used for evaluation.

of width 5 and 7, each group having 2 layers and 96 neurons on each of the layers. The unidirectional convolutional model has 64 filters for each window width from 1 to 4 and 2 convolutional layers as well. Dropout rate was 0.2.

Neural networks are implemented using Keras framework with TensorFlow backend. They are trained with Adam optimizer for at most 50 epochs, training is stopped when the accuracy on development set do not improve for 10 epochs. In case of multitask training the language models are trained for 5 epochs jointly with the main model, batches for different tasks are sampled in random order. The size of mini-batch is 32 for all the runs.

4.2 Results.

Our first evaluation scores the basic model on datasets from Kann et al. (2018) and Grönroos et al. (2019). We compare our with their seq2seq model, the CRF model of Ruokolainen et al. (2013) and the semi-supervised neural model (the one of Kann et al. (2018) using autoencoding and the one of Grönroos et al. (2019) trained with Harris features). The supervised CRF model is retrained by ourselves, while other scores except our own are taken from the original papers. We report two metrics, micro-averaged (per morpheme boundary) boundary F1[7] and word accuracy, which is the fraction of correctly segmented words. All our scores are averaged over 5 independent runs with different random initialization, the standard error is also reported.

Analyzing the results in Table 2, we see that our basic model always outperforms sequence-to-sequence model by a substantial margin, also being ahead of conditional random fields on 4 datasets of 5. That answers our first question: convolutional neural networks seem to work better than other approaches supervised morpheme segmentation even in extremely low-resource setting. In Table 3 we present the scores for our unidirectional model both in its supervised version and in the semi-supervised one, which is trained using multitask learning. We observe that unidirectional convolutions work better than the traditional ones and the multitask training imporves the scores slightly more further.

We conclude that on the mentioned datasets our model outperforms other tested approaches, setting a new state-of-the-art score for them. We also note that one-side CNNs are better than the basic ones, though they have 4 times more parameters. However, basic CNNs of comparable size do not perform better than the smaller ones due to severe overfitting. Gains from semi-supervised training are the more substantial the more data we have, thus the effect on Mexicanero and Yorem Nokki with less than 1000 unlabeled words is the most modest.

5 Conclusion and future work.

We demonstrate that convolutional neural networks outperform other segmentation models in low-resource setting. We argue that this is due to their ability to capture local dependencies, while morpheme segmentation is essentially local by its nature. A similar observation on sentence-level tasks was made in Yin et al. (2017) which demonstrated that CNNs perform better in tasks like answer selection that do not envolve long-distance relations. However, the claims made on 6 languages (5 of the present article and Russian in Sorokin and Kravtsova (2018) and Bolshakova and Sapin (2019)), 4 of which belong to the same family certainly need further proof on other languages and datasets. However, we note that CNNs are (arguably) more effective not only in terms of performance quality, but also in terms of training complexity.

Nonetheless promising, our results still leave a

[7] The work of Kann et al. (2018) reports macro-averaged one, therefore we do not present their boundary F1. We think the micro-averaged version better reflects algorithm properties since the impact of words with larger number of morphemes is higher.

	Word accuracy				Boundary F1			
Language	CNN(our)	seq2seq	CRF	semi-sup	CNN(our)	seq2seq	CRF	semi-sup
Mexicanero	79,4 (0,4)	75,0	78,3	**80,5**	89,7(0,3)	NA	89,2	NA
Nahuatl	59,9 (1,0)	55,9	**64,4**	60,3	77,4(1,0)	NA	80,4	NA
Wixarika	61,4 (0,6)	57,5	58,6	**61,9**	88,2(0,5)	NA	87,8	NA
Yorem Nokki	69,2(0,7)	65,7	65,9	**71,0**	82,6(0,7)	NA	80,3	NA
North Sámi	**71,6**(0,8)	69,1	70,9	71,1	80,8(0,9)	NA	80,0	NA

Table 2: Results of our basic CNN segmentation model in comparison with sequence-to-sequence model (seq2seq), conditional random fields (CRF) and semi-supervised extension of seq2seq (semi-sup). Seq2seq and semi-supervised results for Yuto-Aztecan languages are from Kann et al. (2018), for North Sámi from Grönroos et al. (2019).

	Convolutional (our)			Other	
Language	basic	one-side	one-side+LM	best semi-sup	best
Mexicanero	79,4(0,4)	**80,6**(1,3)	80,1(1,6)	80,5	80,5
Nahuatl	59,9(1,0)	62,8(0,6)	**64,4**(1,1)	60,3	**64,4**
Wixarika	61,4(0,6)	62,9(1,5)	**64,8**(1,1)	61,9	61,0
Yorem Nokki	69,2(0,7)	70,5(0,9)	**71,7**(0,9)	71,0	71,0
North Sámi	71,6(0,8)	72,0(0,5)	**72,5**(0,3)	71,1	71,1

Table 3: Results of our extended CNN models in comparison with the basic one and state-of-the-art. Results for Yuto-Aztecan languages are from Kann et al. (2018), for North Sámi from Grönroos et al. (2019).

huge room for improvement. First of all, the absolute numbers are quite low, only less than two thirds of the words are segmented correctly. The first thing to study is the learning curve of neural segmentation algorithm: it is not so important that a model achieves 60% accuracy on 1000 annotated words, more important is whether it may reach 80% given another thousand of training examples. Another open direction is the incorporation of linguistic features, such as Harris-like distributional measures used in Ruokolainen et al. (2014) or intra-segment interactions regulated by adaptor grammars.

Sometimes morpheme segmentation also requires normalization of morphemes (e.g *studied* ↦ *study* + *ed*). This task is not that straightforward to address with CNNs since the problem is no more reduced to sequence labeling. This is exactly the case for Semitic languages, where morpheme segmentation often depends not only from the word itself, but from wider context (Zeldes, 2018). Since neural networks can work with input vectors of any origin, CNN models have the potential for these tasks also and we hope to address some of these questions in future research.

Acknowledgements

The research was conducted under support of National Technological Initiative Foundation and Sberbank of Russia. Project identifier 0000000007417F630002. The author also thanks Manuel Mager for providing the datasets and help with their processing.

References

Nikolai Dmitrievich Andreev, editor. 1965. *Statictical and combinatorial language modelling (Statistiko-kombinatornoe modelirovanie iazykov, in Russian)*. Nauka.

Elena Bolshakova and Alexander Sapin. 2019. Improving neural morphological tagging using language models. In *Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference "Dialogue*, pages 105–113.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, et al. 2017. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. *arXiv preprint arXiv:1706.09031*.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the*

ACL-02 workshop on Morphological and phonological learning-Volume 6, pages 21–30. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):3.

Ramy Eskander, Owen Rambow, and Smaranda Muresan. 2018. Automatically tailoring unsupervised morphological segmentation to the language. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 78–83.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2019. North sámi morphological segmentation with low-resource semi-supervised sequence labeling. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 15–26.

Zellig S Harris. 1970. Morpheme boundaries within words: Report on a computer test. In *Papers in Structural and Transformational Linguistics*, pages 68–77. Springer.

Martin Haspelmath, Matthew S Dryer, David Gil, and Bernard Comrie. 2005. The world atlas of language structures.

Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *Advances in neural information processing systems*, pages 641–648.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 961–967.

Katharina Kann, Manuel Mager, Ivan Meza-Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. *arXiv preprint arXiv:1804.06024*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Manuel Mager, Elisabeth Mager, Alfonso Medina-Urrea, Ivan Meza, and Katharina Kann. 2018. Lost in translation: Analysis of information loss during machine translation between polysynthetic and fusional languages. *arXiv preprint arXiv:1807.00286*.

Andrew Matteson, Chanhee Lee, Young-Bum Kim, and Heuiseok Lim. 2018. Rich character-level information for korean morphological analysis and part-of-speech tagging. *arXiv preprint arXiv:1806.10771*.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, et al. 2014. Painless semi-supervised morphological segmentation using conditional random fields. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 84–89.

Tatyana Ruzsics and Tanja Samardzic. 2017. Neural sequence-to-sequence learning of internal word structure. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 184–194.

Yan Shao. 2017. Cross-lingual word segmentation and morpheme segmentation as sequence labelling. *arXiv preprint arXiv:1709.03756*.

Kairit Sirts and Sharon Goldwater. 2013. Minimally-supervised morphological segmentation using adaptor grammars. *Transactions of the Association for Computational Linguistics*, 1:255–266.

Alexey Sorokin and Anastasia Kravtsova. 2018. Deep convolutional networks for supervised morpheme segmentation of russian language. In *Conference on Artificial Intelligence and Natural Language*, pages 3–10. Springer.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Linlin Wang, Zhu Cao, Yu Xia, and Gerard de Melo. 2016. Morphological segmentation with window lstm neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Amir Zeldes. 2018. A characterwise windowed approach to hebrew morphological segmentation. *arXiv preprint arXiv:1808.07214*.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21.