An Extensive Empirical Evaluation of Character-Based Morphological Tagging for 14 Languages

Conference Paper · January 2017

DOI: 10.18653/v1/E17-1048

3 authors:

Georg Heigold
Google Inc.
46 PUBLICATIONS **1,742** CITATIONS
SEE PROFILE

Günter Neumann
Deutsches Forschungszentrum für Künstliche Intelligenz
168 PUBLICATIONS **1,847** CITATIONS
SEE PROFILE

Josef van Genabith
German Research Center for Artificial Intelligence DFKI
304 PUBLICATIONS **3,403** CITATIONS
SEE PROFILE

An Extensive Empirical Evaluation of Character-Based Morphological Tagging for 14 Languages

Georg Heigold
DFKI & Saarland University
Saarbrücken, Germany
georg.heigold@dfki.de

Günter Neumann
DFKI
Saarbrücken, Germany
neumann@dfki.de

Josef van Genabith
DFKI & Saarland University
Saarbrücken, Germany
josef.van_genabith@dfki.de

Abstract

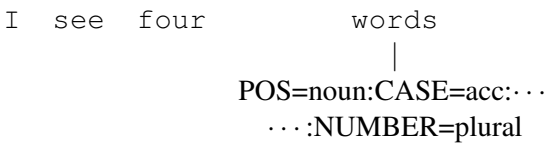
This paper investigates neural character-based morphological tagging for languages with complex morphology and large tag sets. Character-based approaches are attractive as they can handle rarely- and unseen words gracefully. We evaluate on 14 languages and observe consistent gains over a state-of-the-art morphological tagger across all languages except for English and French, where we match the state-of-the-art. We compare two architectures for computing character-based word vectors using recurrent (RNN) and convolutional (CNN) nets. We show that the CNN based approach performs slightly worse and less consistently than the RNN based approach. Small but systematic gains are observed when combining the two architectures by ensembling.

1 Introduction

Character-based approaches have been studied for many applications in natural language processing, including part-of-speech (POS) tagging (dos Santos and Zadrozny, 2014; Ling et al., 2015; Gillick et al., 2016; Plank et al., 2016; Ma and Hovy, 2016), morphological tagging (Labeau et al., 2015), parsing (Ballesteros et al., 2015), named entity recognition (Gillick et al., 2016), language modeling (Ling et al., 2015; Kim et al., 2016), and neural machine translation (Costa-jussà and Fonollosa, 2016). Character-based representations have the advantage of gracefully handling rare or unseen words and tend to produce more compact models as the number of atomic units, i.e., characters, is smaller compared to the number of words in word-level approaches. The issue of rare or unseen words is particularly pronounced when working on morphologically-rich languages, small amounts of training data or noisy user input.

Morphological tagging is the task of assigning a morphological analysis to a token in context. The morphological analysis for a word consists of a sequence of feature:value pairs describing, for example, case, gender, person and tense. A particular concatenation of such feature:value pairs is referred to as a single tag (Oflazer and İlker Kuroz, 1994; Hajic and Hladka, 1998; Mueller et al., 2013).

Following (Müller and Schuetze, 2015), we also add the part-of-speech to this morphological tag and refer to it as POS-MORPH:



Given a word in context, we predict a POS-MORPH tag as a complete unit, rather than as the individual component parts. This approach allows us to share large parts of the model but can only produce POS-MORPH analyses attested in the training data (cf. Table 2). This is still the standard approach to morphological tagging and disambiguation as, given sufficient amounts of training data, the number of POS-MORPH descriptions that cannot be produced usually is small.

Character-based POS tagging (rather than full POS-MORPH tagging) has been extensively evaluated in the literature (dos Santos and Zadrozny, 2014; Ling et al., 2015; Gillick et al., 2016; Plank et al., 2016). The results are competitive but do not systematically outperform the state of the art. Only Plank et al. (2016) report consistent gains by using shallow neural network architectures in combination with multitask learning, multilingual

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 505–513,
Valencia, Spain, April 3-7, 2017.

learning, and pre-trained word embeddings.

State-of-the-art results for morphological tagging (full POS-MORPH tagging) can be found in (Mueller et al., 2013; Müller and Schuetze, 2015). To the best of our knowledge, there has not been much research on character-based morphological tagging so far. Labeau et al. (2015) is an exception but report results for German only. Their best results are on a par with state-of-the-art results. Heigold et al. (2016) show clear gains of character-based over state-of-the-art morphological taggers. However, the evaluation is limited to German and Czech.

Research on character-based approaches in general NLP clearly divides into papers that use CNN-based architectures (dos Santos and Zadrozny, 2014; Kim et al., 2016; Costa-jussà and Fonollosa, 2016) and papers that use LSTM-based architectures (Labeau et al., 2015; Ling et al., 2015; Gillick et al., 2016; Ballesteros et al., 2015; Plank et al., 2016; Ma and Hovy, 2016). There are a number of examples where an LSTM paper reports results of a CNN paper for comparison, such as (Ling et al., 2015) (POS tagging for English) and (Gillick et al., 2016) (named entity recognition for English). However, there is no direct comparison between CNN and LSTM based architectures in morphological tagging.

In this paper, we investigate character-based morphological tagging in more depth. More specifically, the contributions of this paper include:

- the evaluation of character-based morphological tagging on 14 different languages of different morphological complexity;
- the empirical comparison of long-short term memory (LSTM) and convolutional neural network (CNN) based architectures;
- the demonstration of systematic gains of our character-based, language-agnostic morphological tagger over a state-of-the-art morphological tagger across morphologically rich languages; moreover, and perhaps as expected, we show that the relative gains are clearly correlated with the amount of the training data;
- the evaluation of the complementarity of LSTM- and CNN-based architectures by ensemble experiments.

The remainder of the paper is organized as follows. Section 2 summarizes the character-based neural network approaches used in this paper. The data sets and model configurations are described in Section 3 and in Section 4, respectively. The empirical evaluation is presented in Section 5. Section 6 concludes the paper. The Appendix contains a listing of all experimental results obtained in this paper.

2 Character-based Tagging

We assume an input sentence w_1^N with (complex POS-MORPH morphological) output tags t_1^N and a zeroth-order Markov model

$$p(t_1^N|w_1^N) = \prod_{n=1}^{N} p(t_n|w_1^N) \quad (1)$$

whose factors are modeled by a suitable neural network. For character-based tagging, we use the character representation of the word, $w = c_1^M$. This assumes that the segmentation of the sentence into words is known, which is straightforward for the languages under consideration.

At the top level, each input word maps to one complex POS-MORPH morphological output tag. Hence, we can model the position-wise probabilities $p(t|w_1^N)$ with recurrent neural networks, such as long short-term memory recurrent neural networks (LSTMs) (Graves, 2012). Fig. 1 (a) shows such a network architecture where the inputs are the word vectors v_1^N. At the lower level, we use a CNN-based (Fig. 1 (b)) or an LSTM-based (Fig. 1 (c)) architecture to compute the character-based word vectors. As we are using bidirectional LSTMs (BLSTMs) at the top level, we shall refer to the complete architectures as CNNHighway-BLSTM and LSTM-BLSTM. The two architectures are fairly similar. In our opinion, however, there is an important difference between the two. CNNHighway is more constructive in the sense that it explicitly specifies the possible character context widths with a hard upper bound and defines an embedding size for each context width. LSTMs are more generic as they are claimed to implicitly learn these details (Schmidhuber, 1992).

The weights of the network, θ, are jointly estimated using conditional log-likelihood

$$F(\theta) = -\sum_{n=1}^{N} \log p_\theta(t_n|w_1^N). \quad (2)$$

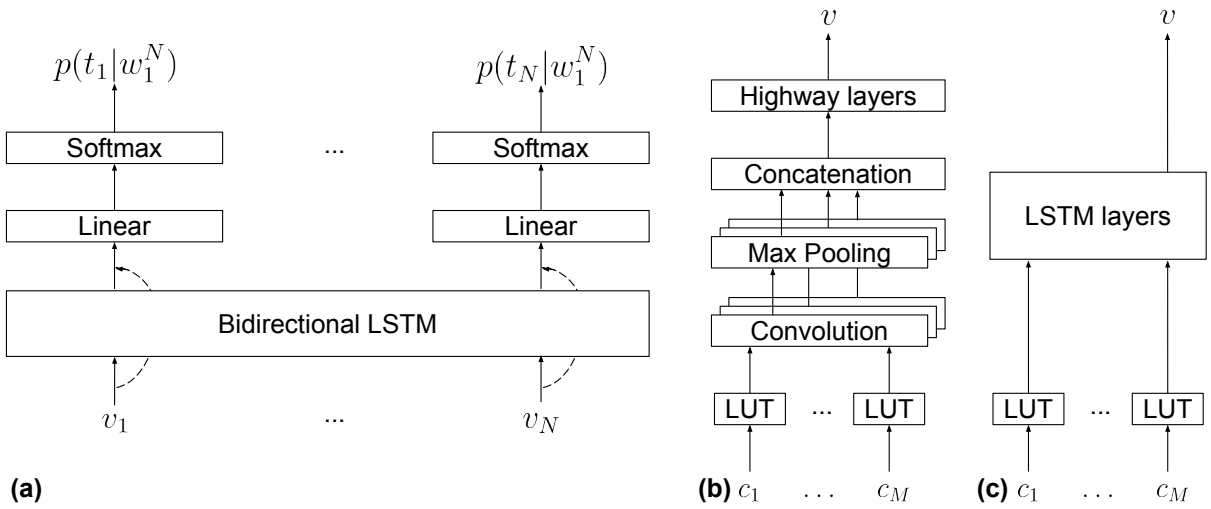


Figure 1: Character-based neural tagging architecture: (a) sub-network mapping word vectors v_1^N to tags t_1^N, dashed arrows indicate optional skip connections, (b) CNNs with different filter widths followed by fully-connected layers with highway connections (CNNHighway), and (c) deep LSTM using the last output to map the character string to a word vector. The networks in (b) and (c) are cloned to produce the input word vectors v_1^N in (a). LUT stands for lookup table.

Learning in recurrent or very deep neural networks is non-trivial and skip/shortcut connections have been proposed to improve the learning of such networks (Pascanu et al., 2014; He et al., 2016). We use such connections (dashed arrows in Fig. 1) for LSTM-BLSTM to alleviate potential learning issues.

At test time, the predicted tag sequence is the tag sequence that maximizes the conditional probability $p(t_1^N|w_1^N)$. For the factorization in Eq. (1), the search can be done position-wise. This significantly reduces the computational and implementation complexity compared to first-order Markov models as used in (Collobert et al., 2011; dos Santos and Zadrozny, 2014; Labeau et al., 2015).

3 Data

Most of the data sets are taken from the UD treebanks[1]. We also use a number of older data sets in order to compare our results with existing results in the literature, including Czech/PDT[2], German/TIGER[3], and Korean/SPMRL[4]. The corpus statistics for the different languages can be found in Table 1. The chosen languages are from different language families: Balto-Slavic (Bulgarian, Czech, Russian), Finnic (Estonian, Finnish), Finno-Ugric (Hungarian), Germanic (German), Indo-Iranian (Hindi), Koreanic (Korean), Romance (Romanian, Semitic (Arabic), and Turkic (Turkish). They include several examples for both agglutinative and fusional languages. The amount of training data ranges from 33k training tokens (Hungarian/UD) to 1,174k training tokens (Czech/UD).

Table 2 summarizes the tag statistics for the different languages. The number of tags is the number of POS-MORPH tags occurring in the training data. We give the test entropy based on a unigram tag model estimated on the training data as a simple measure for the difficulty of the associated sequence classification problem. The type/token ratio (TTR), also known as vocabulary size divided by text length, is computed on 1M words from randomly selected sentences from a different data set[5] and is a simple measure to quantify the morphological complexity of a language (Bane, 2008). A higher TTR value indicates higher morphological complexity.

[1] http://dependencies.org/

[2] https://ufal.mff.cuni.cz/pdt3.0

[3] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html

[4] http://dokufarm.phil.hhu.de/spmrl2013/?animal=spmrl2013

[5] The sentences are all taken from the Wiki dumps on http://linguatools.org/tools/corpora/wikipedia-monolingual-corpora/ and https://archive.org/details/wikipediadumps?&sort=-downloads&page=2.

Table 1: Corpus statistics, OOV≥5 denotes the percentage of test word tokens with five or more occurrences in the training data

Language	Train sentences (k)	Train tokens (k)	Test tokens (k)	OOV≥5 (%)
Arabic/UD	6	256	32	20.7
Bulgarian/UD	9	124	16	27.3
Czech/PDT	39	691	93	17.5
UD	68	1174	174	15.7
English/UD	13	205	25	16.7
Estonian/UD	15	188	24	32.2
Finnish/UD	12	163	9	38.9
French/UD	15	367	7	12.7
German/TIGER	40	760	92	17.2
Hindi/UD	13	281	35	10.1
Hungarian/UD	1	33	4	48.0
Korean/SPMRL	23	296	28	42.7
Romanian/UD	5	109	18	27.6
Russian/UD	47	815	108	19.5
Turkish/UD	4	42	9	46.5

Table 2: Tag statistics, TTR stands for Type/Token Ratio

Language	#Tags	Entropy	TTR (%)
Arabic/UD	320	32.5	12
Bulgarian/UD	448	49.5	12
Czech/PDT	878	77.7	11
UD	1418	97.7	11
English/UD	119	27.9	7
Estonian/UD	787	57.3	13
Finnish/UD	1593	76.1	17
French/UD	197	34.1	8
German/TIGER	681	97.7	13
Hindi/UD	922	56.9	7
Hungarian/UD	652	64.5	14
Korean/SPMRL	1976	119.4	20
Romanian/UD	444	65.8	7
Russian/UD	434	54.6	16
Turkish/UD	987	73.0	10

4 Setups

We use the same model setups for LSTM-BLSTM and CNNHighway-BLSTM as in (Heigold et al., 2016). The hyper-parameters are set to

- CNNHighway: the large setup from (Kim et al., 2016), i.e., character vector size = 15, filter widths ranging from one to seven, number of filters as a function of the filter width $\min\{200, 50 \cdot \text{filter width}\}$, two highway layers
- LSTM: character vector size = 128, two layers with 1024 and 256 nodes

The BLSTM modeling the context of words in a sentence (Fig. 1 (a)) consists of two hidden layers, each with 256 hidden nodes.

These hyper-parameters were tuned on the German TIGER development data and are optimal on a "best effort basis." German is good for the hyper-parameter tuning as it is a relatively hard task (see Table 2) and shows morphological effects both within and across words. Furthermore, the TIGER corpus is relatively large, which reduces statistical fluctuations in training and testing. Apart from these considerations, the choice was random. Furthermore, we tested language-specific tuning for a few languages, but it does not seem to give further gains. Moreover, the network hyper-parameters were tuned to give best accuracy rather than most compact models or even comparable numbers of

parameters as our application is not constrained by memory or runtime. The hyper-parameters were then used for all languages. We ran the external tools MarMoT[6] and JNN[7] (see Appendix) with the suggested default values.

The networks are optimized as described in (Heigold et al., 2016). In particular, the optimization is done with RMSProp (Tieleman and Hinton, 2012), with a fixed initial learning rate and a learning rate decay of two every tenth epoch for German, TIGER, and is adjusted for the other languages according to the amount of training data. The batch size is always 16. Furthermore, we use dropout. The dropout probability is empirically set to 0.4 for Hungarian and Turkish, which only have a very limited amount of training data (Table 1), and to 0.2 for all other languages.

5 Empirical Evaluation

We empirically evaluate an LSTM-based and a CNN-based architecture for character-based morphological tagging (Section 2) and compare them against MarMoT, a state-of-the-art morphological tagger (Mueller et al., 2013). For the evaluation we use twelve different morphologically-rich languages with different characteristics, plus two morphologically-poor languages for contrastive results (Section 3). The configurations are described in Section 4.

Fig. 2 plots the relative gain over MarMoT (see Appendix A for more details) against the amount of training data. The horizontal dotted line at 0% indicates the MarMoT baseline. The blue squares are for LSTM-BLSTM results. Connecting them for the morphologically-rich languages shows a clear, nearly-linear dependency of the relative gain on the amount of training data. Only the data point for Turkish at 40% is an outlier (should be around 20%). This result suggests that compared to MarMoT, LSTM-BLSTM is very data efficient. Even for very small amounts of training data (e.g., 33k tokens for Hungarian), the relative gain is still 15%. On the other hand, more data helps. In case of Czech, increasing the amount of training data from 691k (Czech/PDT) to 1174k (Czech/UD) tokens leads to some additional gain and yields almost a 50% relative gain. It should be noted, however, that the two data sets use different tag sets, with the Czech/UD one being more complex than the Czech/PDT (Table 2).

We use an LSTM-BLSTM of the same size for all languages, although the amount of training data varies by roughly two orders of magnitude. Therefore, it is a valid question if a larger model specifically designed for Czech/UD or a smaller model for Turkish/UD would improve the results. We have developed locally tuned and tested larger and smaller models in terms of number of nodes or layers but with similar or worse performance: -0.1% with more nodes (Czech) or approx. -1% with fewer nodes or fewer layers (Turkish). This observation suggests that the configuration optimized for German is fairly robust across many different languages, which is an attractive property from a practical perspective.

In contrast, we do not observe a gain of LSTM-BLSTM over MarMoT for English and French. Both languages are considered to be morphologically poor, as supported by the tag statistics in Table 2. This may be because of the low morphological complexity, i.e., a character representation does not add much information to a word representation. Another explanation might be that the linguistic experts have focused on English and French in the last decades and found a good set of features, which however does not well generalize to other, morphologically more complex languages.

It is tempting to analyze these results in more detail by splitting languages into sub-categories. Here, we refrain from doing so as it is delicate to draw conclusions from very small sample sizes (3-4 languages, say).

The green circles (in Fig. 2) are for CNNHighway-BLSTM results, a neural network architecture that has been developed for character-based language modeling (Kim et al., 2016). Overall, LSTM-BLSTM and CNNHighway-BLSTM perform similarly, see Fig. 2. Looking at the details, however, CNNHighway-BLSTM tends to perform slightly worse and less consistently than LSTM-BLSTM.

While LSTM-BLSTM and CNNHighway-BLSTM perform similarly they may capture complementary effects. To measure the complementarity of the two architectures, we build an ensemble consisting of the LSTM-BLSTM and the CNNHighway-BLSTM by taking the geometric mean of the scores. The accuracies are shown in Fig. 2 as LSTM+CNNHighway-BLSTM. Ex-

[6] http://cistern.cis.lmu.de/marmot/
[7] https://github.com/wlin12/JNN

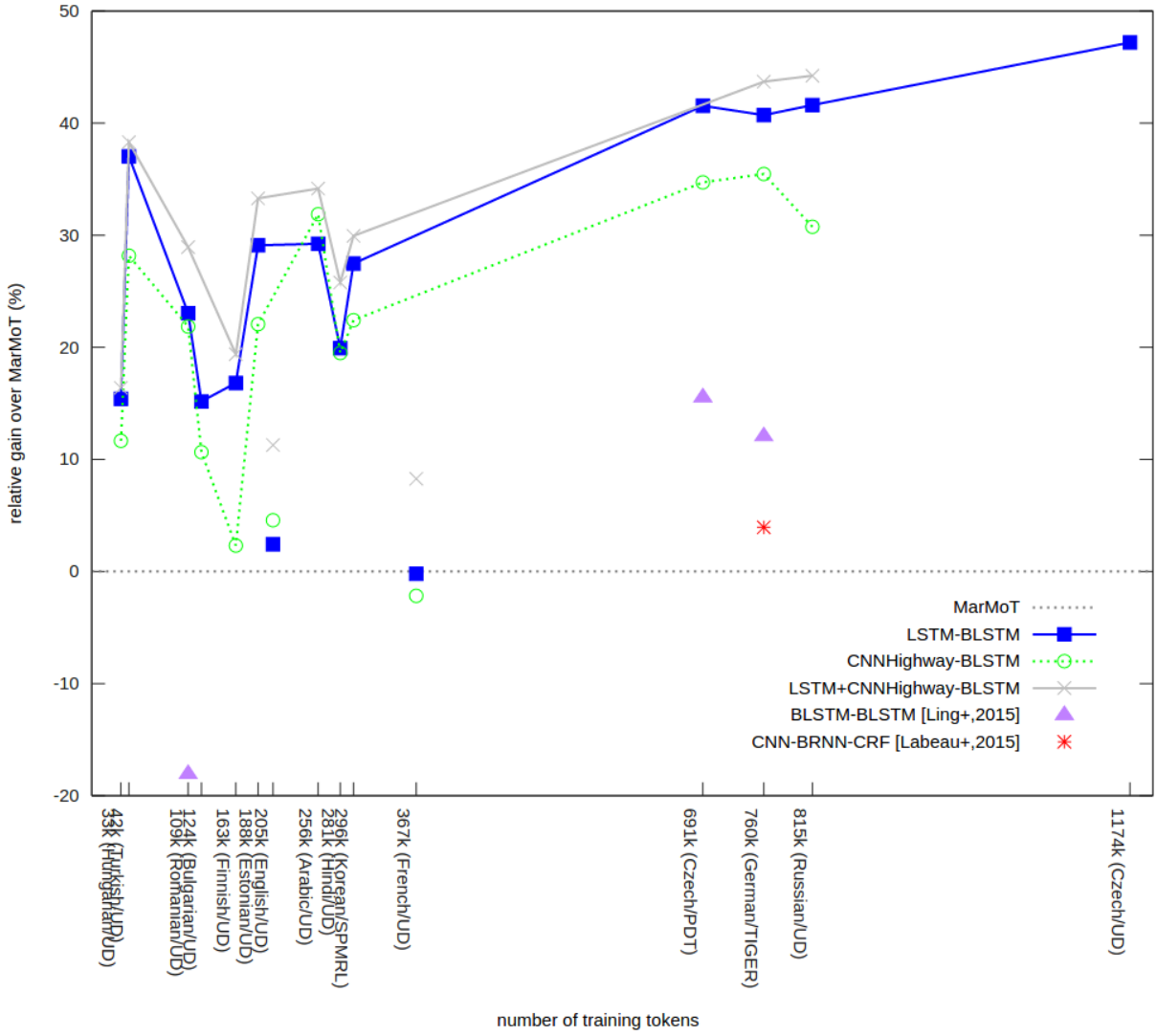


Figure 2: Relative gains (%) over MarMoT

cept maybe for English and French, we observe marginal but consistent gains over LSTM-BLSTM or CNNHighway-BLSTM.

For additional comparison, we add a few additional points in the plot. The red cross indicates the result from (Labeau et al., 2015), which is a combination of a CNN, a bidirectional RNN, and a Markov model. The purple triangles are generated with the external tool JNN[8], which implements a shallow BLSTM-BLSTM (i.e., only one bidirectional LSTM layer in each BLSTM). One might expect that this model performs better on smaller data sets. But actually, it is clearly worse both for large (Czech/PDT and German/TIGER) and small data sets (Romanian/UD).

6 Summary & Future Work

In this paper, we demonstrated that a character-based neural approach can achieve consistent improvements over a state-of-the-art morphological tagger (MarMoT). The evaluation included a dozen of languages of different morphological complexity and with different characteristics. The relative gains for the morphologically-rich languages range from 15% to almost 50%, with a clear dependency on the amount of training data. Several aspects are remarkable about this result.

First, these results use the same model architecture with the same number of layers and nodes, without any language-specific modifications. Further local language and training data setting specific tuning does not seem to help much.

Second, the neural approach seems to be more data efficient than the baseline tagger with manually designed features, also when only 30k training tokens are available.

Third, a fairly generic deep and hierarchical recurrent neural network architecture seems to perform as well or better than a more specialized convolutional neural network based architecture.

Fourth, to keep the setup as simple as possible, we have not used advanced techniques which are reported to lead to improvements, including a non-trivial structured prediction model (e.g., a first-order Markov model) (Collobert et al., 2011; dos Santos and Zadrozny, 2014; Labeau et al., 2015; Ma and Hovy, 2016), additional unsupervised data (e.g., via word2vec) (Müller and Schuetze, 2015; Ling et al., 2015; Plank et al., 2016; Ma and Hovy, 2016), combination of different word representations (Labeau et al., 2015; Ma and Hovy, 2016; Plank et al., 2016), multilingual learning (Gillick et al., 2016; Plank et al., 2016), and auxiliary tasks (Plank et al., 2016). Future work will include the investigation of these more advanced techniques. From this perspective, our paper provides a baseline for future research in multilingual character-based neural morphological tagging.

Last but not least, we do not observe any gains for English and French (except when using ensembles). This may be due to the low morphological complexity of these languages or because manual feature engineering has focused on these languages over the last decades with good results.

[8] https://github.com/wlin12/JNN

Acknowledgment

This work has been partly funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 645452 (QT21).

References

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 349–359, Lisbon, Portugal, September. Association for Computational Linguistics.

Max Bane. 2008. Quantifying and measuring morphological complexity. In C.B. Chang and H.J. Haynie, editors, *Proceedings of the 26th West Coast Conference on Formal Linguistics*, pages 69–76. Cascadilla Proceedings Project, Somerville, MA, USA.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany, August. Association for Computational Linguistics.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California, June. Association for Computational Linguistics.

Alex Graves. 2012. *Supervised sequence labelling with recurrent neural networks*. Studies in Computational Intelligence. Springer, Heidelberg, New York.

Jan Hajic and Barbora Hladka. 1998. Tagging inlective languages: Prediction of morphological categories for a rich structured tagset. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 483–490, Montreal, Quebec, Canada, August. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2016. Neural morphological tagging from characters for morphologically rich languages. *CoRR*, abs/1606.06640.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *AAAI*, Phoenix, AZ, USA, February.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen. 2015. Non-lexical neural architecture for fine-grained pos tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 232–237, Lisbon, Portugal, September. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.

Thomas Müller and Hinrich Schuetze. 2015. Robust morphological tagging with word representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 526–536, Denver, Colorado, May–June. Association for Computational Linguistics.

Kemal Oflazer and İlker Kuroz. 1994. Tagging and morphological disambiguation of Turkish text. In *Proceedings of the Applied natural language processing*.

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. How to construct deep recurrent neural networks. In *ICLR*.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany, August. Association for Computational Linguistics.

Cícero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *ICML*, Beijing, China, June.

Jürgen Schmidhuber. 1992. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

A Raw Results

This appendix contains Table 3 with the raw results used in this paper. When available, the best comparable error rates from the literature are used. Otherwise, we produced the error rates with the publicly available tools and the suggested default values. More specifically, we used the state-of-the-art tagger MarMoT[9] for the baselines and the LSTM-based POS tagger JNN[10] for some contrastive results.

[9] `http://cistern.cis.lmu.de/marmot/`

[10] https://github.com/wlin12/JNN

Table 3: Tag error rates (%) on test sets, some of which are taken from the literature: (a) (Mueller et al., 2013), (b) (Labeau et al., 2015)

Language	MarMoT[9]	CNN -biRNN-CRF	BLSTM -BLSTM[10]	LSTM -BLSTM	CNNHighway -BLSTM
Arabic/UD	9.13			6.46	6.22
Bulgarian/UD	5.73			4.86	5.12
Czech/PDT	7.46[a]		6.30	4.36	4.87
UD	6.97			3.68	
English/UD	7.00			6.83	6.68
Estonian/UD	8.11			5.75	6.32
Finnish/UD	7.79			6.48	7.61
French/UD	5.08			5.09	5.19
German/TIGER	11.42[a]	10.97[b]	10.04	6.77	7.37
Hindi/UD	11.44			9.16	9.21
Hungarian/UD	26.49			22.41	23.40
Korean/SPMRL	18.60			13.49	14.43
Romanian/UD	7.64		9.02	5.88	5.97
Russian/UD	6.08			3.55	4.21
Turkish/UD	17.28			10.88	12.41