Sub-label dependencies for Neural Morphological Tagging – The Joint Submission of University of Colorado and University of Helsinki for VarDial 2018

Miikka Silfverberg
Department of Linguistics
University of Colorado Boulder
mpsilfve@iki.fi

Senka Drobac
Department of Modern Languages
University of Helsinki
senka.drobac@helsinki.fi

Abstract

This paper presents the submission of the UH&CU team (Joint University of Colorado and University of Helsinki team) for the VarDial 2018 shared task on morphosyntactic tagging of Croatian, Slovenian and Serbian tweets. Our system is a bidirectional LSTM tagger which emits tags as character sequences using an LSTM generator in order to be able to handle unknown tags and combinations of several tags for one token which occur in the shared task data sets. To the best of our knowledge, using an LSTM generator is a novel approach. The system delivers sizable improvements of more than 6%-points over a baseline trigram tagger. Overall, the performance of our system is quite even for all three languages.

1 Introduction

This paper[1] presents the joint submission of University of Colorado and University of Helsinki for the 2018 VarDial shared task on morphosyntactic tagging of Croatian, Serbian and Slovenian tweets (Zampieri et al., 2018). Morphosyntactic tagging is a useful preprocessing task when parsing morphologically complex languages since these typically encode syntactic information as inflectional material in word forms. For example, both of the following Croatian words forms are inflected forms of 'dog': **pas** and **psa**. However, the first one is far more likely to encode a grammatical subject since it displays nominative case. This demonstrates that coarse POS tags are not sufficient for capturing all syntactically relevant aspects of words in morphologically complex languages. Instead, rich morphological tags are needed.

It is not sufficient to train one morphosyntactic tagger and expect it to perform well in all domains. The reason for this is that the performance of data-driven models typically suffers when they are applied to domains which considerably differ from their training domain. Consequently, NLP models often deliver poor results when applied to the social media domain since most models are trained on newswire or related, more formal, domains. This is a problem because text analysis for social media has become increasingly important both from an economical and research perspective in recent years.

Social media differs from newswire in many respects. As explained in Section 3, Croatian, Serbian and Slovenian text in the social media domain often lacks diacritics, which ordinarily are a prominent feature in the orthographies of these languages. Moreover, orthographic rules concerning capitalization are frequently ignored. Furthermore, our error analysis in Section 5 shows that Twitter text contains a large amount of foreign, mainly English, loan words. These are some of the reasons why NLP systems can fail to deliver good performance on social media text.

The VarDial shared task specifically targets the social media domain. Our system is trained on collections of morphosyntactically annotated Tweets (Ljubešić et al., 2017a; Ljubešić et al., 2017b; Erjavec et al., 2015). It is an LSTM (Hochreiter and Schmidhuber, 1997) morphosyntactic tagger which utilizes pretrained subword-aware word embeddings and character-based word embeddings. The setup is shown in Figure 1. This basic setup for an LSTM tagger is not new. However, our system does have novel aspects, especially in the output layer.

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 37–45
Santa Fe, New Mexico, USA, August 20, 2018.

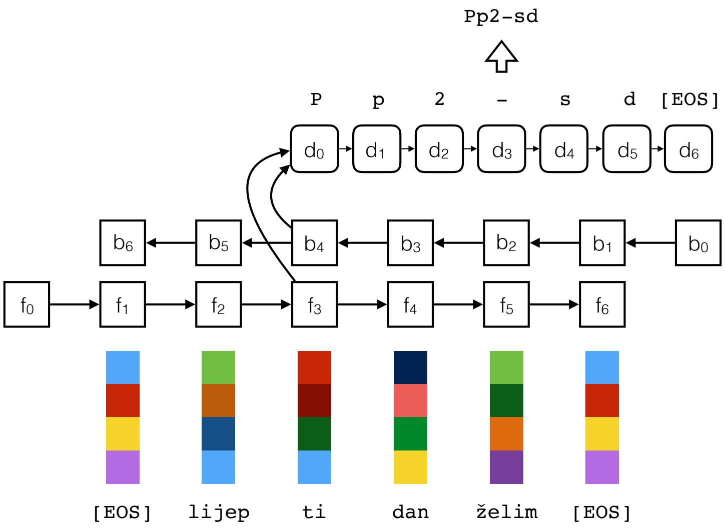

Figure 1: Our system starts by padding the input sentence $x_1, ..., x_T$ with end-of-sequence symbols `[EOS]`. It then embeds all tokens in the sentence (see Figure 3 for details). Embeddings are fed into a bidirectional LSTM encoder which outputs forward and backward representation vectors f_t and b_{T-t} at each position t. To predict the morphosyntactic tag at position t, the concatenated representation $[f_t; b_{T-t}]$ is then fed into an LSTM generator which emits the MULTEXT-East tag for token x_t one character at a time.

Previous approaches to neural morphosyntactic tagging have either treated complex morphological tags like `Npmsn` (the singular nominative of a masculine noun) as atomic units or predicted each feature (for example `N`, `p`, `m`, `s`, `n`) separately. Both of these approaches are insufficient for our needs. The first approach is suboptimal because the system will treat MULTEXT-East tags (Erjavec, 2012) `Npmsn` and `Npmsl` as completely separate entities even though both are in fact proper noun tags which share number and gender. Indeed, Müller et al. (2013) and Silfverberg et al. (2014) show that sub-tag dependencies improve the performance of linear taggers. It is conceivable that the same applies to deeper neural architectures.

i	Cc
jel	Var3s Qq
izljubio	Vmp-sm
punicu	Ncfsa
?	Z

Figure 2: In this sentence 'and did he kiss mother in law?', the token **jel** receives two separate tags because it is a contraction of the verb **je** ('to be' in 3rd person singular) and the interrogative particle **li**.

The second approach, namely individually predicting each feature of the tag, does take into account individual sub-tags. However, this approach does not model their dependencies or the complete tag in any way which also seems problematic. In the case of morphosyntactic tagging in the MULTEXT-East schema, there is also a more serious problem with predicting each sub-label in isolation. Namely, tokens can sometimes receive multiple tags, for example in the case of contractions (Ljubešić et al., 2017a). An example of this is shown in Figure 2. A straightforward approach to predicting sub-tags cannot handle this situation. Therefore, we opt for using an LSTM generator for emitting tags. It can model both individual sub-tags and dependencies between sub-tags. Because LSTM networks excel at long range dependencies, there is reason to believe that our approach also captures information about complete MULTEXT-East tags.

The paper is structured in the following way: In Section 2 we present related approaches neural to morphosyntactic tagging and tagging in the social media domain. In Section 3, we present our LSTM tagger. Section 4 presents the data sets used in the VarDial task and Section 5 presents our experiments and results. Finally, we present discussion and directions for future work in Section 6 and conclude the paper in Section 7.

2 Related Work

Our system is inspired by the neural POS tagger introduced by Dozat et al. (2017), however, we have extended their approach to handle morphological tagging. In the past two years, POS tagging for morphologically complex languages has received a fair amount of attention. Starting with the work by Plank et al. (2016a), neural approaches, particularly bidirectional LSTM taggers, have dominated the field. This is exemplified by the entry of Dozat et al. (2017) for the 2017 CoNLL shared task on multilingual parsing, where their neural POS tagger delivered the best results by far for nearly all languages (Zeman et al., 2017).

Even though work on neural POS tagging has received more attention, there are a number of papers on neural morphosyntactic tagging. Heigold et al. (2016a) evaluate several architectures for morphosyntactic tagging[2] of German and Czech. They find that pretrained word embeddings bring large gains in presence of small training sets and that character-based architectures deliver the best performance. Heigold et al. (2016b) extend these experiments to 12 additional languages.

Most existing systems for morphosyntactic tagging treat complex morphosyntactic tags in the same way as POS tags, that is, they do not model the internal structure of tags. As an exception to this, Krasnowska-Kieraś (2017) predict each sub-tag in complex morphosyntactic tags separately. As mentioned above, this is not a sufficient solution in our case since it does not address the problem of multiple morphosyntactic possible tags for one token. Therefore, we opt for using an LSTM generator for emitting tags. To the best of our knowledge, this approach is novel.

We utilize automatically tagged data from the web domain (Ljubešić and Klubička, 2014) to improve the performance of our system. Plank and Nissim (2016b) use a similar approach for POS tagging of Italian tweets. They use automatically tagged data from the social media domain and find that it can deliver sizable improvements. Our results point in the same direction.

3 Methods

This section describes our bidirectional LSTM tagger. It also describes how automatically tagged web data is used for improving tagger accuracy and the data transformations that we perform on the web data in order improve performance in the Twitter domain.

3.1 A Neural Morphological tagger

Our system is an unstructured morphosyntactic LSTM tagger[3]. We utilize character-based embeddings and pretrained embeddings and the system emits morphological tags using an LSTM generator. This allows us to both emit tags, which we have not seen in the training data, and emit combinations of several tags for one token. This is necessary for handling contractions present in the shared task datasets, as explained above.

Embedding layer Our word embedding layer combines three types of word embeddings: pretrained word embeddings, randomly initialized word embeddings and character-based embeddings. See Figure 3 for a visualization.

Pretrained embeddings are initialized using FastText (Bojanowski et al., 2017) which treats word forms as a bags of character n-grams. We use FastText because it can provide an embedding vector both for tokens that were observed during training and for other tokens. This is important when dealing with morphologically complex languages, where out-of-vocabulary (OOV) rates are typically high. We train pretrained FastText embeddings using large quantities of plain text. In addition to pretrained embeddings, we use regular randomly initialized token based embeddings. It is common practice to include both types of embeddings in a tagger.

[2] Many authors including (Heigold et al., 2016b) refer to morphosyntactic tagging as morphological tagging.

[3] The term unstructured refers to the fact that the tag for each token in the sentence is predicted in isolation. This is common practice in the field of neural morphological tagging.

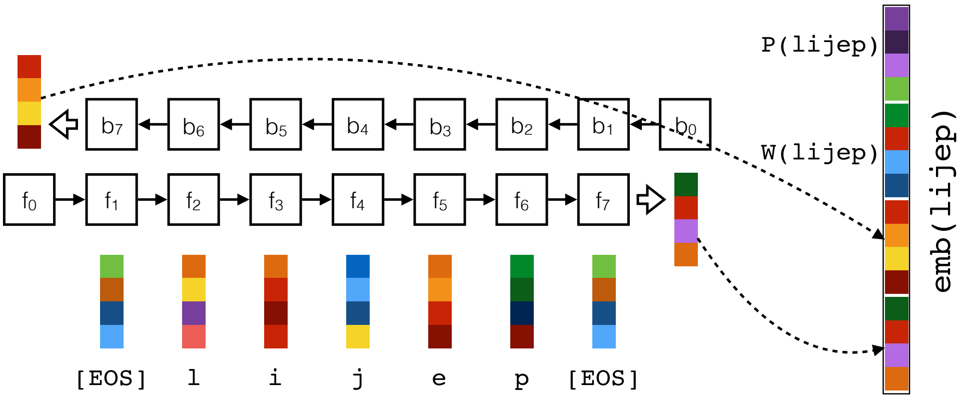

Figure 3: Word embeddings in our system are a concatenation of four vectors: a pretrained word embedding $\mathrm{P}(w)$, a randomly initialized whole word embedding $\mathrm{W}(w)$ as well as forward and backward character based embeddings $\mathbf{f_C}(w)$ and $\mathbf{b_C}(w)$ computed by a bidirectional LSTM encoder.

Finally we use character-based embeddings based on a bidirectional character-level LSTM encoder. To compute character-level embeddings, we treat the input word as a sequence of characters $c_1, ..., c_N$ and pad it with end-of-sequence symbols resulting in a sequence $c_0, ..., c_{N+1}$. We then compute character embeddings $\mathrm{E}(c_i)$ for each character in the sequence $c_0, ..., c_{N+1}$. Subsequently, we use the forward component of the LSTM encoder for encoding the sequence $\mathrm{E}(c_0), ..., \mathrm{E}(c_{N+1})$ into a representation vector. Similarly, we use the backward component of the encoder for encoding the reverse sequence $\mathrm{E}(c_{N+1}), ..., \mathrm{E}(c_0)$ into a representation vector. We use the final cell-state $[\mathbf{f_C}; \mathbf{b_C}]$ of the bidirectional LSTM encoder as the representation of the sequence.

As a final step, we concatenate all vectors into a unified token representation. As stated above, any word form, whether seen during training or not, will receive a pretrained embedding vector. Therefore, we do not need to treat OOV tokens differently with regard to the pretrained embedding. In contrast, the random initialized token embedding may encounter unknown tokens during test time. Therefore, we use a special unknown word token [UNK], whose embedding is initialized randomly. During training, we then replace input token embeddings with the embedding for [UNK] with probability $p_{\texttt{WORDUNK}}$. In order to simulate the distribution of OOV tokens, [UNK] embeddings are trained exclusively on input tokens which occur once in the training data.

It may also happen that we encounter unknown characters in the test data. Therefore, we also use an unknown character symbol and train it analogously to the unknown word symbol, that is, we randomly replace character with [UNK] during training with probability $p_{\texttt{CHARUNK}}$.

Sentence-level LSTM encoder We use a bidirectional encoder LSTM for deriving a sequence of state vectors $[\mathbf{f_t}; \mathbf{b_{T-t}}]$ from token representation vectors. The state vector $\mathbf{s_t} = [\mathbf{f_t}; \mathbf{b_{T-t}}]$ is the concatenation of the cell states of the forward and backward components of the bidirectional LSTM.

Tag generator The sentence-level representation at position t is fed into an LSTM generator, which generates MULTEXT-East tags, for example Npmsn, as character sequences. Formally, the generator is a recursive function $\mathrm{G}(\mathrm{E}_{\mathrm{TAG}}(c_{k-1}), \mathbf{h}_k, \mathbf{s_t})$ conditioned on the embedding $\mathrm{E}_{\mathrm{TAG}}(c_{k-1})$ of the previously generated character c_{k-1}, the current hidden state of the generator $\mathbf{h_k}$, and the hidden state of the sentence-level encoder LSTM $\mathbf{s_t}$. The function value $\mathrm{G}(\mathrm{E}_{\mathrm{TAG}}(c_{k-1}), \mathbf{h}_k, \mathbf{s_t})$ is a distribution over possible characters occurring in MULTEXT-East tags and the output character c_k is determined as the mode of that distribution. The process is initialized by setting c_0 to an end-of-sequence symbol [EOS]. We apply teacher forcing (Goldberg, 2017) when training the generator.

3.2 Data transformation

There are a number of differences between the language use in Croatian, Serbian and Slovenian Tweets and language use in more formal domains. Capitalization and diacritics are often omitted in Tweets and there are far more English and foreign language words in Tweets. Additionally, there are English words whose orthography has been adapted to Croatian, Serbian or Slovenian spelling. Although, addressing all of these points is likely to improve tagging accuracy, we decided to focus on capitalization and diacritics.

Before training pretrained embeddings, we remove all diacritics from words in the embedding training data. For example, **korištenjem** → **koristenjem**. When indexing the word embedding, we remove diacritics from the query word. In addition, we make embeddings case insensitive so **grijanje**, **Grijanje** and **GRIJANJE** all receive the same embedding vector.

3.3 Improving Performance using Web Data

As further explained in Section 5, we use automatically tagged Croatian, Serbian and Slovenian web data (Ljubešić and Klubička, 2014) for improving the performance of the tagger. This is done simply by combining web data and Twitter data into one training corpus. We train several taggers combining different parts of the web data with the Twitter data and perform majority voting to get the final result.

3.4 Implementation Details

We use 300 dimensional randomly initialized word embeddings, pretrained word embeddings, character embeddings and sub-tag embeddings. We use 2-layer bidirectional LSTM encoders for computing character representations, sentence-level representations and for generating output tags.

Due to the small size of the Twitter training corpora and the well known tendency of deep learning models to overfit, we add Gaussian noise with standard deviation 0.2% to randomly initialized word embeddings and pretrained embeddings during training. We also apply 50% dropout to the parameters of all LSTM networks. Additionally, we replace characters with an `[UNK]` symbol with probability 0.1 during training. We also replace the randomly initialized word embedding for words that occur once in the training corpus with an `[UNK]` embedding with probability 30%. During training, we use minibatches of size 50 and train for 100 epochs using Adam (Kingma and Ba, 2014). The tagger is implemented using DyNet (Neubig et al., 2017).

4 Data

P	Attribute (en)	Value (en)	Code (en)
0	CATEGORY	Noun	N
1	Type	common	c
		proper	p
2	Gender	masculine	m
		feminine	f
		neuter	n
3	Number	singular	s
		plural	p
4	Case	nominative	n
		genitive	g
		dative	d
		accusative	a
		vocative	v
		locative	l
		instrumental	i
5	Animate	no	n
		yes	y

Table 1: Croatian and Serbian Specifications for Noun

For training we use the following data sets: Twitter data for Croatian (Ljubešić et al., 2017a), Serbian (Ljubešić et al., 2017b) and Slovenian (Erjavec et al., 2015). Additionally, we use automatically tagged web data for all thee languages (Ljubešić and Klubička, 2014). All data sets were tagged according to MULTEXT-East Morphosyntactic Specifications[4]. For all three languages the specification recognizes 12 parts of speech (Noun, Verb, Adjective, Pronoun, Adverb, Adposition, Conjunction, Numeral, Particle, Interjection, Abbreviation, Residual) and each category has different numbers of language specific attributes and values. For example, in all three languages, nouns have 5 attributes (Type, Gender, Number, Case and Animate) and each attribute has language-specific values. Table 1 shows attributes and corresponding values for the Croatian and Serbian Noun category. Slovenian has the same attributes, but different values for the category Number (singular, plural and dual) and Case (there is no vocative in Slovenian). Two examples for noun tags are shown in figure 4.

```
Uskrs   Npmsn (Noun, proper, masculine, singular, nominative)
sudbinu Ncfsa (Noun, common, feminine, singular, accusative)
```

Figure 4: Tag explanation for two nouns: "**Uskrs**" (eng. *Easter*) and "**sudbinu**" (eng. *destiny*)

As mentioned above, the MULTEXT-East specification allows for several tags for one token. This happens for contractions like **jel** which are in fact combinations of two ore more distinct word form **je** "to be" and **li**, which is an interrogative particle, in this case.

5 Experiments and Results

We perform experiments on morphosyntactic tagging of Croatian, Serbian and Slovenian Twitter data. Additionally, we use automatically tagged web data for each language to improve performance.[5] For each language, we create pretrained FastText embeddings using the first 10M sentences from the web data.[6] We then form ten tagger training sets using Twitter training data and the tagged web data. These training sets are used for training ten models, in total. Each training set contains the entire original Twitter data but all of them contain disjoint segments of the web data. We use 500K tokens of web data for each training set (more than this degraded results in preliminary experiments). The web data segments are consecutive 500K token chunks data starting at the top of the data set.

	CROATIAN			SERBIAN			SLOVENIAN		
	POS	MOR	TAG	POS	MOR	TAG	POS	MOR	TAG
BASELINE	-	-	0.834	-	-	0.832	-	-	0.832
OUR SYSTEM	0.943	0.886	0.887	0.957	0.900	0.900	0.946	0.884	0.884

Table 2: Accuracy for part-of-speech (POS), morphological features (MOR) and the complete morphosyntactic tag (TAG).

We apply the ten different systems for tagging the test set and perform majority voting to get the final test set tag for each word. We compare the system against a baseline HunPos trigram tagger (Halácsy et al., 2007) which is described in Zampieri et al. (2018). The results are shown in Table 2. Our system substantially outperforms the baseline on all three languages.

Table 3 shows the most common tagging errors that our system makes. As can be seen, confusions between noun tags like `Npmsn` and the foreign word tag `Xf` are frequent for all three languages. Most of these concern English words which the tagger incorrectly identifies as Croatian, Serbian or Slovenian words and labels accordingly. Another common error type is that nominatives are tagged as accusatives or vice versa. For example, many Croatian and Slovenian singular masculine nominatives `Ncmsn` are incorrectly tagged as inanimate accusatives `Ncmsan`. This is an understandable error since Croatian and Slovenian do not overtly mark singular accusatives of inanimate nouns (Barić et al., 1995; Pauliny et al., 1968).

In Serbian, a common error type is the confusion of conjunctions `Cs` and adverbs `Rgp`. This happens because adverbs can be used for combining sentences into one. If one of the sentences is subordinate

[4] http://nl.ijs.si/ME/V4/msd/html/

[5] We do not use the manually tagged out-of-domain data provided in the shared task.

[6] We do not use the morphosyntactic tags for pretrained embeddings. We use default settings for FastText.

CROATIAN			SERBIAN			Slovenian		
No	GT	Result	No	GT	Result	No	GT	Result
69	Npmsn	Xf	56	Rgp	Cs	44	Xf	Npmsn
45	Ncmsn	Ncmsan	43	Cs	Rgp	36	Ncmsn	Ncmsan
38	Qo	Cc	42	Ncmsn	Ncmsan	28	Sl	Sa
30	Xf	Npmsn	35	Qo	Cc	28	Ncmsan	Ncmsn
29	Ncmsan	Ncmsn	35	Ncmsan	Ncmsn	27	Q	Px------y
28	Npmsn	Ncmsn	30	Vmm2s	Vmr3s	27	Npmsn	Ncmsn
23	Ncmsn	Xf	27	Npmsn	Xf	25	Ncmsn	Npmsn
22	Sl	Sa	26	Agpnsny Rgp	Xf	24	Agpnsn	Rgp
20	Xf	Ncmsn	22	Cc	Qo	23	Xf	Ncmsn
20	Sa	Sl	19	Sl	Sa	22	Npfsn	Ncfsn

Table 3: Most common mistakes per line (total error lines: HR 2428, SR 2325, SL 2246)

to the other one, the word is analyzed as an adverb. However, when neither sentence is subordinate, the word is analyzed as a conjunction.This, of course, requires rather elaborate analysis of the sentences involved. Therefore, it is not surprising that the distinction results in tagging-errors. The same confusion happens in the Croatian test set, however it is less common. The reason for this is that tokens that are causing this error (**kad(a)** ’when’, **kako** ’how’ and **gde** ’where’, cro. ’gdje’) are almost two times more often present in the Serbian test set than in Croatian as shown in Table 4.

	sr	hr
kad (when)	112	72
kada (when)	53	13
kako (how)	66	40
gde/gdje (where)	24	13
TOTAL	255	138

Table 4: Occurrences of words *kad*, *kada*, *kako* and *gde* (Croatian *gdje*) in Serbian and Croatian test data

6 Discussion and Future Work

Our results show that neural methods deliver large improvements in accuracy compared to a traditional trigram tagger. This is a nice result because our neural tagger is unstructured whereas the HunPos baseline is a second order structured model. However, it is not as easy to beat a well engineered discriminative model as the shared task results show (Zampieri et al., 2018).

Our error analysis uncovers a number of directions for future work. It would clearly be beneficial to be able to better model foreign words. Also better contextual modeling is required in order to be able to distinguish nominatives and accusatives in cases where there is no overt morphological marking. It will probably be quite challenging to model the distinction between conjunctions and adverbs in Serbian since this may require rather deep analysis of the embedded sentences.

It is possible that the errors related to the confusion between noun forms as well as conjunctions and adverbs are related to data sparsity, on one hand, and tagging errors in the web data, on the other hand. This requires further analysis.

7 Conclusions

In this paper, we presented an LSTM tagger for morphosyntactic tagging of Croatian, Serbian and Slovenian tweets. The tagger employs pretrained FastText embeddings and an LSTM generator for emitting tags. Our experiments show that a neural approach results in large improvements compared to a traditional trigram tagger. However, our error analysis still uncovers a number of directions for future work.

Especially better modeling of foreign words could help to further improve results.

Acknowledgements

We wish to thank the anonymous reviewers for their valuable feedback. The first author is supported by the Society of Swedish Literature in Finland.

References

Eugenija Barić, Mijo Lončarić, Dragica Malić, Slavko Pavešić, Mirko Peti, Vesna Zečević, Marija Znika, et al. 1995. *Hrvatska gramatika*. Školska knjiga.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.

Tomaž Erjavec, Nikola Ljubešić, and Nataša Logar. 2015. The slWaC Corpus of the Slovene Web. *Informatica*, 39(1):35–42.

Tomaž Erjavec. 2012. Multext-east: Morphosyntactic resources for central and eastern european languages. *Lang. Resour. Eval.*, 46(1):131–142, March.

Yoav Goldberg. 2017. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 209–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

G. Heigold, J. van Genabith, and G. Neumann. 2016b. Scaling character-based morphological tagging to fourteen languages. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3895–3902, Dec.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Katarzyna Krasnowska-Kieraś. 2017. Morphosyntactic disambiguation for Polish with bi-LSTM neural networks. In Zygmunt Vetulani and Patrick Paroubek, editors, *Proceedings of the 8th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 367–371, Poznań, Poland. Fundacja Uniwersytetu im. Adama Mickiewicza w Poznaniu.

Nikola Ljubešić and Filip Klubička. 2014. {bs,hr,sr}wac - web corpora of bosnian, croatian and serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 29–35. Association for Computational Linguistics.

Nikola Ljubešić, Tomaž Erjavec, Maja Miličević, and Tanja Samardžić. 2017a. Croatian twitter training corpus ReLDI-NormTagNER-hr 2.0. Slovenian language resource repository CLARIN.SI.

Nikola Ljubešić, Tomaž Erjavec, Maja Miličević, and Tanja Samardžić. 2017b. Serbian twitter training corpus ReLDI-NormTagNER-sr 2.0. Slovenian language resource repository CLARIN.SI.

Thomas Müller, 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.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Eugen Pauliny, Jozef Ružička, and Jozef Štolc. 1968. *Slovenská gramatika*. SPN.

Barbara Plank and Malvina Nissim. 2016b. When silver glitters more than gold: Bootstrapping an italian part-of-speech tagger for twitter. *CoRR*, abs/1611.03057.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016a. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Miikka Silfverberg, Teemu Ruokolainen, Krister Lindén, and Mikko Kurimo. 2014. Part-of-speech tagging using conditional random fields: Exploiting sub-label dependencies for improved accuracy. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 259–264. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shuon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de-Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada, August. Association for Computational Linguistics.