Tagger for Polish Computer Mediated Communication Texts

Wiktor Walentynowicz
Wrocław University
of Science and Technology

Maciej Piasecki
Wrocław University
of Science and Technology

Marcin Oleksy
Wrocław University
of Science and Technology

{wiktor.walentynowicz,maciej.piasecki,marcin.oleksy}@pwr.edu.pl

Abstract

In this paper we present a morpho-syntactic tagger dedicated to Computer-mediated Communication texts in Polish. Its construction is based on an expanded RNN-based neural network adapted to the work on noisy texts. Among several techniques, the tagger utilises fastText embedding vectors, sequential character embedding vectors, and Brown clustering for the coarse-grained representation of sentence structures. In addition a set of manually written rules was proposed for post-processing. The system was trained to disambiguate descriptions of words in relation to Parts of Speech tags together with the full morphological information in terms of values for the different grammatical categories. We present also evaluation of several model variants on the gold standard annotated CMC data, comparison to the state-of-the-art taggers for Polish and error analysis. The proposed tagger shows significantly better results in this domain and demonstrates the viability of adaptation.

1 Introduction

Morpho-syntactic disambiguation (called also morpho-syntactic tagging) is an important pre-processing step in many text processing pipelines (e.g. terminology and information extraction), especially in the case of highly inflected languages where tagging is tightly correlated with lemmatisation. Work on the development of programs for morpho-syntactic disambiguaters, henceforth *taggers*, has been concentrated on texts written in the standard language, i.e. containing only small percentage free of jargon, extra-linguistic elements like codes or symbols, etc. Computer-mediated communication (CMC) texts, especially texts from different kinds of social media, are written in language that often is significantly different from the standard language, see Sec. 3. While in the case of texts in a standard language, a state-of-the-art tagger reaches near 95% of accuracy in disambiguation, taggers for CMC texts express much lower accuracy. Moreover, much fewer works are devoted to tagging CMC texts than text of the standard language, while the problem of processing CMC texts is continuously growing in importance and numbers of applications. Thus, the fast growth of social media requires appropriate adaptation, or even expansion, of the tagging methods, to serve growing interest and requirements in the processing of texts of such kinds. According to our best knowledge, a morpho-syntactic tagger dedicated to CMC texts in the Polish language has not been yet developed or at least made publicly available. The goal of this work is a far going adaptation of a tagger for the standard Polish to the demands of CMC texts in Polish.

In the rest of the paper, first solutions for tagging the standard Polish language and CMC texts in general are discussed. Next, we analyse characteristic features of CMC texts on the basis of a collected CMC corpus. A tagger model is proposed. Finally, we presented evaluation of the tagger and discuss perspectives for its further development.

2 Related Works

The construction of our CMC tagger has been inspired by a tagger for the Polish language that won the PolEval competition (Kobyliński and Ogrodniczuk, 2017) in 2017 that is called *Toygger* (Krasnowska-Kieraś, 2017). *Toygger* is based on a recursive neural network – a bidirectional LSTM. Recursive layers extract features based on an input

Proceedings of Recent Advances in Natural Language Processing, pages 1295–1303,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_148

vector encoding morphological information and including embedded vectors. The extracted features are next transferred to separated non-linear layers, where each represents a different part of the morpho-syntactic tag. In Sec. 4 we propose several expansions to this model which are aimed at providing better handling of noisy texts. Aside from Toygger, another tagger – *KRNNT* (Wróbel, 2017), also based on a bidirectional LSTM network, achieved very good results in the PolEval contest. The main difference between these two is in the encoding of the input text and features generated for it. Toygger uses text encoded as a sequence of embedding vectors in combination with information from the morphological analysis performed with the morpho-syntactic analyser *Morfeusz2* (Kieraś and Woliński, 2017), and KRNNT uses information from the morphological analysis from Maca (Radziszewski and Śniatowski, 2011) (that is also employing *Morfeusz* inside) in combination with predetermined features based on a set of features from the *Concraft* (Waszczuk, 2012) tool.

In a similar way, complex systems such as dependency parser *COMBO* (Rybak and Wróblewska, 2018), in their internal taggers began to use LSTM networks with similar architecture to the stand-alone taggers mentioned above. Inside the COMBO system the tagger component obtains the features extracted from a bidirectional LSTM layer and passes them through a fully connected network with one hidden layer with a softmax activation function. That network predicts a universal part-of-speech tag and a tagset specific tag (for instance a grammatical class in the case of a tagset for Polish). The morphological features have similar networks for each feature type.

Apart from the systems created for the Polish language, several solutions were proposed for other Slavic languages or even highly inflected languages in general. The winning solution of the competition organised at the VarDial 2018 conference (Zampieri et al., 2018) was a system based on a bidirectional LSTM network, which instead of classifying words with morpho-syntactic tags generates the tags in a character per character way and uses a different type of a recursive network (Silfverberg and Drobac, 2018) for this purpose. The way the network is trained ensures that a tag that has never occurred in the training data is not generated. Such a manner of getting replies from networks allows also words, which are concatenated together from several morphemes, to have a 'multi-part' tag, i.e. a cluster of several tags *per se*. This solution was inspired by the Sequence-to-Sequence architecture. It also shows the positive effect of combining embedding vectors for words and characters, which is confirmed by other works from this genre, such as (Plank et al., 2016). Another work in this field is (Ljubešić, 2018), which compares a tagger model based on Conditional Random Fields with a model based on a recursive neural network in disambiguation of Slovene, Serbian and Croatian CMC texts. The differences between the two models are small (about 0.02), which leads to the conclusion that both methods are worth considering in further research. However, a comparative study in (Östling, 2018) shows that better results are achieved with good manual processing of features than with extending and deepening the architecture.

Apart from CRF-based models, the other methods presented here do not pay attention to both sides of the context. Using a bidirectional LSTM layers does bring information about the context, but only before the word (or after the word when looking from the other direction) and the method of combining the results of these two directions does not guarantee focusing on both sides of context in the same degree. Therefore, our solution proposes to add to the network information about the context from the Brown clustering algorithm. Here, we were inspired by another work about tagger adaptation (Ljubešić et al., 2017).

3 Computer-Mediated Communication Corpus

Computer-mediated communication (CMC) texts are part of user-generated content (UGC) data. Due to the nature of this kind of data, the texts commonly include many mistakes and problematic phenomena. A linguistic analysis of the data as well as the results of similar research (see e.g (Pluwak et al., 2016)) helped us to define distinctive features, which are related to various text levels such as: notation (e.g. lack of diacritics, spelling mistakes, typos, omissions of capital letters, incorrectly connected or disconnected segments, lack of or poor punctuation), morphology and syntax (e.g. incorrect word endings, token repetition, lack of phrase elements), or lexical issues (e.g. emojis and special characters, internet slang, characters replacements, abbreviated forms, URL addresses, hashtags, mentions). Most of these phenomena require specific solutions in annotation guidelines. Since the available data in Polish is based on the sources of a different nature – they are mostly edited and officially published texts (Przepiórkowski et al., 2012) – there was a need to create a CMC corpus manually annotated with morphological information.

The *Corpus of the Colloquial Polish Language*[1] (CCPL) used in all experiments presented in this paper consists of 7,561 documents (402,810 tokens). All the source texts were posted by users on online social media platforms, so they have the characteristics of user-generated content (UGC). The texts include opinions, tweets, comments, social media posts and chat utterances.

The whole corpus was manually annotated with morphological information and next morphosyntactically disambiguated by the team of professional linguists from the Wrocław University of Science and Technology. The inter-annotator agreement for various pairs of annotators was calculated. The results ranged from 0.91 to 0.97.

4 Tagger Model

4.1 Data and Preprocessing

National Corpus of Polish (NCP) as the basic training data, and only in some experiments we expanded the training data with annotated texts from social media coming from CCPL, see Section 5. This subcorpus of NCP contains a bit above 1.2 million of manually annotated and disambiguated tokens from different sources. It is commonly used as training-testing data set for a tagging task in Polish language. For the test data set we used CCPL, which is described in Section 5.1.

We have decided not to apply text normalisation before tagging process in order to save as much information as possible from the text structure which can be helpful in next stages of processing like sentiment recognition. Many of the methods proposed in literature are based on a process: normalisation followed by tagger application. Thus, a comparison of our solution with them is difficult, as the text tagged is different. Therefore we do not handle segmentation problems in our tagging system. To deal with typos and lack of diacritics we focus on character representation together with suffix representation and choose to use the *fastText* embedding (Bojanowski et al., 2017), because it is based on the n-gram based subword word representation. In addition, we obtained cluster information to get better representation of contexts for similar words. We applied the Brown Clustering algorithm (Brown et al., 1992) to group words on the basis of the one million subcorpus of NCP (Przepiórkowski et al., 2012) (this is the same data set which is used as training data). The Brown Clustering is a method of hierarchical clustering of words based on their contexts. We assumed that words belonging to the same clusters have the same probability of the contextual occurrence under the condition of the occurrence of preceding and following words. In principle, we want to achieve good results in spite of processing noisy textual data, due to the knowledge of their structure on the coarse-grained description level based on Brown clusters of words. We assume that such representation helps to determine to which group of words an unknown or broken word may belong to. Other elements of CMC texts are emoticons, URL and e-mail addresses, hashtags and user mentions. We handle these cases with handwritten rules.

In several experiments, see Section 5, the evaluation of the whole tagging process was performed on the entire CCPL corpus, because it was not involved in training in those cases.

4.2 Input Text Representation

In order to improve the tagger ability to generalise over the training data, we used a representation based on distributional vector models. An input vector for a word from the processed sentence is constructed as a concatenation of several subvectors representing different properties of a word:

1. a morphological information vector,
2. a suffix character embedding vector,
3. a suffix index in the set of known suffixes,
4. a suffix embedding vector,
5. a word embedding vector from a *fastText*-based model,
6. a whole word character embedding,
7. a Brown cluster embedding vector.

The morphological information vector expresses jointly information collected from all tags

[1] http://hdl.handle.net/11321/637

that are possible for a given token according to the morphological analysis (in the case of unknown words, the full vector is set). Sets of possible tags are represented as a sequence of bits: every single bit of each vector represents a possible grammatical class or a value of some grammatical category (an attribute), e.g. case, number gender etc. For instance for the Polish word *jedzenie* 'food' we obtain a vector with two bits set for the two grammatical classes: *noun* (*jedzenie* as 'food') and *gerund* (*jedzenie* 'eating'), bits for the *nominal* and *accusative* genders, one bit for the *singular* number, one for the *m3* gender etc. The morphological vector is intended to be a kind of regularisation constraining the tagging process and to make the tagger decisions compatible with the morphological analyser.

The suffix character embedding vector – the part 2 – is trained during the time of learning by using an additional small (64 hidden units) biLSTM network to represent suffixes as character sequences. In all experiments, we set the size of a suffix to 3, based on the results published in the (Krasnowska-Kieraś, 2017) (who tested experimentally 4 and 5, too) and our previous experiments. This vector is aimed at recognition of different suffixes (carrying important morphological information) and their similarity.

The suffix index (3) and suffix embedding vector (4) represent also suffixes, but on the level of suffixes as tokens, not sequences of characters. Concerning the former, a set of known suffixes is first extracted from the learning data. Next each recognised suffix is represented by its index during training and testing. In the case of the latter, for suffixes as tokens their vector embeddings are trained during learning the sequential tagging task. During testing and application for each known suffix its embedding vector representation is searched for in the look-up table layer. By introducing these two components we want to emphasise the presences of more frequent suffixes that often express more specific morpho-syntactic information.

The fifth part is a word embedding vector for the whole token obtained from *fastText* model. We use a *fastText* model for Polish that was trained on a very large corpus of Polish (Kocoń, 2018). This vector introduced lexical element to the representation, but due to the nature of the distributional model, words of similar distribution receive similar vectors. Moreover, *fastText* (a subword distributional model) assigns also vectors to unknown words on the basis of n-gram structure.

The sixth part is a character embedding for whole words. Similar to suffix character embedding in part 2, it is learned during the training the whole tagger and has similar architecture. The most significant difference is that it takes whole words at the input, not just suffixes.

The last component (7) is an embedding vector for the Brown cluster of the given word. In a similar way to the suffix embedding vector, vector embeddings for Brown clusters are trained during learning the tagger. If the word does not appear in any cluster, it receives a cluster index for Out of Vocabulary (OOV) words. During testing and application the vectors are read from the look-up table layer. Brown clusters offer a coarse-grained representation of the input sequence that helps to analyse OOV words.

4.3 Network and Processing

The core part of the tagger is a deep neural network. The input vector is a sequence of the combined word vectors. It is sent to two bidirectional LSTM layers with 512 hidden units each. 50% dropout is applied to both LSTM layers. The goal of these layers is to calculate features for each word of the input sentence. Next, these features are used to feed down separated layers. These separated layers are *softmax layers*. The first one is dedicated to the grammatical class and the rest (13) to the 13 different grammatical categories (morphological attributes). The whole network is trained with the help of the *RMSprop optimizer* (Tieleman and Hinton, 2012) and the *Categorical Cross Entropy loss function*.

Due to the variability of CMC texts the main processing by the neural network is supplemented by deterministic post-processing which is performed in three steps:

1. verification of the correctness of a predicted tag,

2. final tag selection,

3. rule-based error detection and correction.

Concerning the first, the correctness of the predicted tag is checked in relation to the set of all possible tags proposed by the morphological analysis. Checking correctness of predicted tag is simple task. On the basis of the predicted grammatical class we verify if the attributes obtain values

number	accuracy
No clusters	85.21%
500	85.46%
750	85.44%
1000	85.44%
1250	85.31%
1500	85.23%
1750	**85.48%**
2000	85.36%

Table 1: Influence of the number of clusters on the tagger strict accuracy.

specified for this grammatical class. In the case of the lack of a tag exactly matching the predicted one among the tags obtained from the morphological analysis, for the final tag selection, we choose a tag that is in the minimal Levenshtein distance of its form to the form of the predicted tag, i.e. the predicted tags are somehow mapped onto tags available from the morphological analysis (in the case of out-of-vocabulary words the full set of tags is assumed as the result of the morphological analysis).

In order to improve an automatic tagging process, we developed and applied several rules. They specify morphological interpretations for selected words directly encountered in text. This concerns, in particular, emojis or internet addresses. In some cases a rule covers only the first characters which serve to identify the relevant word form, e.g. it was not possible to list all URL addresses but the rule using the expression [*if 'https://' in w*] could be applied to all word forms that begin with *https://*).

A morphological interpretation for several word forms (or types of word forms) were assigned irrespective of the interpretation automatically predicted by the tagger. An example of such a general rule is given below:

$if\ 'https://'\ in\ w\ or\ 'http://'\ in\ w:$

$corrected\_tag =' subst:sg:nom:m3'$

The results of the first attempt to error analysis were the basis for drawing up also specific rules for the word forms tagged initially with specific morphological information, e.g.

$if\ (w\ ==\ 'jak'\ or\ w\ ==\ 'tak')$

$and\ predicted\_tag == 'adv:pos':$

$corrected\_tag\ =\ 'adv'$

settings	accuracy
SCE(128) + SE(64)	86.09%
SCE(128) + SE(64) + CLE(64)	*87.06%*
SCE(128) + SE(64) + CLE(64) + R	**87.87%**
CE(128) + SE(64) + CLE(64) + R	87.39%
CE(128) + SCE(128) + CLE(64) + R	87.67%

Table 2: CMC tagger strict accuracy in relation to the different ways of composing the input vector.

5 Evaluation

5.1 Experiments

In our research, we focused on two main aspects

- determining the best number of Brown clusters,
- and selecting the best possible configuration of the input vector components.

In all experiments the input vector included the components representing the morphological analyses and the *fastText* based representation of words. The *fastText* component vector size was fixed to 300 elements.

First, we performed tests to find the best number of the Brown clusters. The results of these experiments are shown in Table 1. The performance of the tagger is measured in the *strict accuracy*, i.e. only the assigned tags that completely, in relation to all its components, match the tag from the manual annotation are treated as correct solutions. The results show that the addition of clustering, regardless of its size, improves the tagging accuracy. Finally, we set the number of clusters to 1,750 in accordance with the best result obtained during the first experiments.

Next, we performed several experiments that were aimed at investigating the influence of the different joint vector components on the tagger accuracy. The results of the most important ones are presented in Table 2 where the shortcuts mean:

SCE – suffix characters embedding,

SE – suffix embedding,

CE – character embedding,

CLE – clustering embedding,

R – postprocessing rules.

The numbers in the round brackets remind about the size of the given vector component. In Table 2 the performance of the tagger is measured, like above, in the *strict accuracy*. The results show that suffix characters embedding brings the most benefits. This is consistent with the intuition, that in the case of words that are blured by noise inside them and OOV words their suffix can tell us most about their morphology. Also, adding rule-based post-processing to the tagger output increased its final accuracy.

On the basis of the experiments, for the final version of the system we chose the input vector consisting of the following components: morphological information, suffix characters embedding, suffix embedding, fastText embedding and clustering embedding for 1,750 clusters.

After selecting the network architecture parameters, we made an additional experiment. Using cross-validation we conducted a test during which we trained the tagger on the two combined data sets, namely: the manually annotated part of NCP and a subset of the manually annotated part of CCPL (i.e. the training folds). The manually annotated CCPL subcorpus was divided into five parts further on referred to as *folds*. The sub-models during the cross-validation process were trained on an NCP with four folds from CCPL and the sub-model was tested on the fifth fold from CCPL. Our tagger tested in this way achieved an average accuracy of 90.14%.

Finally, we compared the version of our CMC Tagger trained on the combined NCP and four folds of CCPL with the two taggers for Polish treated as a baseline, namely:

- *MorphoDiTa-pl* (Piasecki and Walentynowicz, 2017) is accessible at `http://ws.clarin-pl.eu/tager.shtml` and its source code at `https://github.com/ufal/morphodita`.

- *Toygger*, already mentioned, originally trained on NCP with *word2vec* embedding and suffix information feature, with 20 epochs.

The results of the tests done on the folds of CCPL are presented in Table 3.

Tagger	Accuracy (strong)
CMC Tagger	**90.14%**
MorphoDiTa	81.32%
Toygger	86.12%

Table 3: Strong accuracy (identical morphosyntactic class and values of grammatical categories) measured on CCPL corpus.

5.2 Results

We performed detailed linguistic error analysis. It covered the word forms differently tagged by human annotator and morphological tagger. 600 word forms most frequently judged inconsistently (3,675 error instances) were analysed. Among them several error types can be distinguished that mainly correspond to grammatical categories incorrectly recognised and every tuple (word form – human annotator tag – automatically ascribed tag) was assigned to one of the categories presented in Table 4.

The most common error concerned grammatical class. The most frequently confused classes are: adverb – particle-adverb (174 of 3,675 error instances) and coordinating conjunction – particle-adverb (90 instances). The word that appeared to be most difficult to judge was *to* (‘this’, ‘then’, ‘to be’, adverb) (263 error instances), which could be interpreted as adjective, predicative, noun, particle-adverb or subordinating conjunction depending on the context. Tagger had also problems with emojis correct recognition (177 instances).

Generally, the issues described above concern parts of speech that are very often the source of confusion for human annotators. Furthermore, the distribution of tagger errors is very similar to the observed distribution of inconsistencies in manual CCPL tagging. This shows that the mistakes are related to the difficult cases in general. A few use cases from the test dataset are presented below.

In the original, the fragment of sentence looks as follows:

“*[...] a mu sie nie spodoba i po wezystkm.*”

Correctly (without typing errors) this sentence would look like this:

“*[...] a mu się nie spodoba i po wszystkim.*”

(English: “*and he will not like it, and this is all.*”).

The CMC Tagger output for this sentence is shown in Table 5. Our tagger for the word *sie* (reciprocal participle *się* but written without diacritics) wrongly chose adjective as the grammatical class. This problem originated from the morphological analysis. Word form *sie* exists in the dictionary and the tagger could not recognise it

object of inconsistency	percent
number of assigned categories	4.71%
base	6.80%
grammatical class	31.16%
case	14.45%
number	2.29%
gender	22.42%
rection	4.60%
person	1.41%
aspect	0.33%
human error	9.31%
other	2.53%

Table 4: Selected categories of the CMC Tagger errors.

Form	Tag
a	conj
mu	ppron3:sg:dat:m1:ter:nakc:npraep
sie	adj:pl:nom:m2:pos
nie	qub
spodoba	fin:sg:ter:perf
i	conj
po	prep:loc
wezystkm	subst:sg:loc:m1

Table 5: Example sentence number 1

as a corrupted form of *się*, which is a reflexive pronoun or reciprocal participle (an component of a compound verb). The form *wezystkm* (i.e. in its proper form: *wszystkim* ‘everything’/‘all of them(case:dative)’) is an unknown (OOV) word for the morphological analyser, but our tagger made only a small mistake in the gender attribute – it should be neutral.

In the case of a sentence that is grammatically constructed correctly, like the one in Table 6, our tagger works quite well. This sentence in English looks as follows: “In crediting the car purchase all went very smoothly and quite decent repayments.”.

6 Conclusions

We presented that standard approaches to morpho-syntactic disambiguation must be adapted to specific domains of non-standard texts, like CMC and User Generated Content texts, e.g. including social media texts. Thus, we proposed significant expansions to the state-of-the-art tagger for Polish, namely *Toygger*, that resulted in large gain in performance in CMC texts, as measured on the manually annotated gold standard for this domain. In the proposed expansions we focused on the representation and appropriate encoding in the input vector the information about: suffix types, better word embedding models (i.e. taking into account the sub-word level) and word clusters generated on the basis of standard text, but enabling coarse-grained text representation.

Form	Tag
W	prep:loc:nwok
kredytowaniu	ger:sg:loc:n:imperf:aff
zakupu	subst:sg:gen:m3
auta	subst:sg:gen:n
poszło	praet:sg:n:perf
bardzo	adv:pos
sprawnie	adv:pos
i	conj
całkiem	adv
przyzwoite	adj:pl:nom:f:pos
spłaty	subst:pl:nom:f
.	interp

Table 6: Example sentence number 2

In future we want to focus on improving the usage of the morpho-syntactic information, e.g. in a form of partial recognition of possible dependencies. We plan to expand morphological vectors with the representation of the manually written constraints. We also want to work on semi-automatic extraction of contextual post-processing rules for improvement of the tagger performance. At the same time we want research how to expand representation of words on character level to get better recognition of noisy elements in text. An important challenge is handling of the segmentation in a correct way or applying a more advanced normalisation process before tagging. The tagger is available on the open licence from: http://hdl.handle.net/11321/634.

Acknowledgments

Partially funded by: Project "SentiCognitiveServices – next generation service for automating voice of customer and social media support based on artificial intelligence methods" (POIR.01.01.01-00-0806/16), which is co-financed by European Union through European Regional Development Fund under the Smart Growth Programme 2014 – 2020.

References

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. https://doi.org/10.1162/tacl_a_00051.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.* 18(4):467–479. http://dl.acm.org/citation.cfm?id=176313.176316.

Witold Kieraś and Marcin Woliński. 2017. Morfeusz2 - analizator i generator fleksyjny dla języka polskiego. *Język Polski* pages 75–83.

Łukasz Kobyliński and Maciej Ogrodniczuk. 2017. Results of the poleval 2017 competition: part-of-speech tagging shared task. In Zygmunt Vetulani and Patrick Paroubek, editors, *Human language technologies as a challenge for computer science and linguistics: 8th language & technology conference : November 17-19, 2017, Poznań, Poland : proceedings*, Fundacja Uniwersytetu im. Adama Mickiewicza, Poznań, pages 362–366. http://ltc.amu.edu.pl/book/papers/PolEval1-2.pdf.

Jan Kocoń. 2018. KGR10 FastText polish word embeddings. CLARIN-PL digital repository. http://hdl.handle.net/11321/606.

Katarzyna Krasnowska-Kieraś. 2017. Morphosyntactic disambiguation for polish with bi-lstm neural networks. In Zygmunt Vetulani and Patrick Paroubek, editors, *Human language technologies as a challenge for computer science and linguistics: 8th language & technology conference : November 17-19, 2017, Poznań, Poland : proceedings*, Fundacja Uniwersytetu im. Adama Mickiewicza, Poznań, pages 367–371. http://ltc.amu.edu.pl/book/papers/PolEval1-2.pdf.

Nikola Ljubešić. 2018. Comparing CRF and LSTM performance on the task of morphosyntactic tagging of non-standard varieties of south Slavic languages. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pages 156–163. https://www.aclweb.org/anthology/W18-3917.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. 2017. Adapting a state-of-the-art tagger for south Slavic languages to non-standard text. In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*. Association for Computational Linguistics, Valencia, Spain, pages 60–68. https://doi.org/10.18653/v1/W17-1410.

Robert Östling. 2018. Part of speech tagging: Shallow or deep learning? *Northern European Journal of Language Technology (NEJLT)* 5:1–15. http://www.nejlt.ep.liu.se/2018/v5/a01/index.html.

Maciej Piasecki and Wiktor Walentynowicz. 2017. Morphodita-based tagger adapted to the polish language technology pages 377–381. http://ltc.amu.edu.pl/book/papers/PolEval1-2.pdf.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *CoRR* abs/1604.05529. http://arxiv.org/abs/1604.05529.

Agnieszka Pluwak, Wojciech Korczynski, and Marek Kisiel-Dorohinicki. 2016. Adapting a constituency parser to user-generated content in polish opinion mining. *Computer Science (AGH)* 17:23–44.

Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. 2012. *Narodowy Korpus Języka Polskiego [in Polish]*. Wydawnictwo Naukowe PWN. http://nkjp.pl/settings/papers/NKJP_ksiazka.pdf.

Adam Radziszewski and Tomasz Śniatowski. 2011. MACA. CLARIN-PL digital repository, `http://hdl.handle.net/11321/20`. http://hdl.handle.net/11321/20.

Piotr Rybak and Alina Wróblewska. 2018. Semi-supervised neural system for tagging, parsing and lematization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Brussels, Belgium, pages 45–54. https://www.aclweb.org/anthology/K18-2004.

Miikka Silfverberg and Senka Drobac. 2018. Sub-label dependencies for neural morphological tagging – the joint submission of university of colorado and university of Helsinki for VarDial 2018. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pages 37–45. https://www.aclweb.org/anthology/W18-3904.

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(2):26–31.

Jakub Waszczuk. 2012. Harnessing the CRF complexity with domain-specific constraints. The case of morphosyntactic tagging of a highly inflected language. *Proceedings of COLING 2012* pages 2789–2804.

Krzysztof Wróbel. 2017. Krnnt: Polish recurrent neural network tagger. In Zygmunt Vetulani and Patrick Paroubek, editors, *Human language technologies as a challenge for computer science and linguistics: 8th language & technology conference : November 17-19, 2017, Poznań, Poland : proceedings*, Fundacja Uniwersytetu im. Adama Mickiewicza, Poznań, pages 386–391. http://ltc.amu.edu.pl/book/papers/PolEval1-6.pdf.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second vardial evaluation campaign. In Marcos Zampieri, Preslav Nakov, Nikola Ljubešić, Jörg Tiedemann, Shervin Malmasi, and Ahmed Ali, editors, *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, Association for Computational Linguistics, Santa Fe, New Mexico, USA. http://hdl.handle.net/10138/249333.