Char-RNN for Word Stress Detection in East Slavic Languages

Maria Ponomareva
ABBYY
Moscow, Russia
maria.ponomareva@abbyy.com

Kirill Milintsevich
University of Tartu
Tartu, Estonia
kirill.milintsevich@ut.ee

Ekaterina Artemova
National Research University
Higher School of Economics
Moscow, Russia
echernyak@hse.ru

Abstract

We explore how well a sequence labeling approach, namely, recurrent neural network, is suited for the task of resource-poor and POS tagging free word stress detection in the Russian, Ukranian, Belarusian languages. We present new datasets, annotated with the word stress, for the three languages and compare several RNN models trained on three languages and explore possible applications of the transfer learning for the task. We show that it is possible to train a model in a cross-lingual setting and that using additional languages improves the quality of the results.

1 Introduction

It is impossible to describe Russian (and any other East Slavic) word stress with a set of hand-picked rules. While the stress can be fixed at a word base or ending along the whole paradigm, it can also change its position. The word stress detection task is important for text-to-speech solutions and word-level homonymy resolving. Moreover, stress detecting software is in demand among Russian learners.

One of the approaches to solving this problem is a dictionary-based system. It simply keeps all the wordforms and fails at OOV-words. The rule-based approach offers better results; however collecting the word stress patterns is a highly time consuming task. Also, the method cannot manage words without special morpheme markers. As shown in (Ponomareva et al., 2017), even simple deep learning methods easily outperform all the approaches described above.

In this paper we address the following research questions:

1. how well does the sequence labeling approach suit the word stress detection task?
2. among the investigated RNN-based architectures, what is the best one for the task?
3. can a word detection system be trained on one or a combination of languages and successfully used for another language?

To tackle these questions we:

1. compare the investigated RNN-based models for the word stress detection task on a standard dataset in Russian and select the best one;
2. create new data sets in Russian, Ukrainian and Belarusian and conduct a series of mono- and cross-lingual experiments to study the possibility of cross-lingual analysis.

The paper is structured as follows: we start with the description of the datasets created. Next, we present our major approach to the selection of neural network architecture. Finally, we discuss the results and related work.

2 Dataset

In this project, we approach the word stress detection problem for three East Slavic languages: Russian, Ukrainian and Belarusian, which are said to be mutually intelligible to some extent. Our preliminary experiments along with the results of (Ponomareva et al., 2017) show that using context, i.e., left and right words to the word under consideration, is of great help. Hence, such data sources as dictionaries, including Wiktionary, do not satisfy these requirements, because they provide

Proceedings of VarDial, pages 35–41
Minneapolis, MN, June 7, 2019

only single words and do not provide context words.

To our knowledge, there are no corpora, annotated with word stress for Ukrainian and Belarusian, while there are available transcriptions from the speech subcorpus in Russian[1] of Russian National Corpus (RNC) (Grishina, 2003). Due to the lack of necessary corpora, we decided to create them manually.

The approach to data annotation is quite simple: we adopt texts from Universal Dependencies project and use provided tokenization and POS-tags, conduct simple filtering and use a crowdsourcing platform, Yandex.Toloka[2], for the actual annotation.

To be more precise, we took Russian, Ukrainian and Belarusian treebanks from Universal Dependencies project. We split each text from these treebanks in word trigrams and filtered out unnecessary trigrams, where center words correspond to NUM, PUNCT, and other non-word tokens. The next step is to create annotation tasks for the crowdsourcing platform. We formulate word stress annotation task as a multiple choice task: given a trigram, the annotator has to choose the word stress position in the central word by choosing one of the answer options. Each answer option is the central word, where one of the vowels is capitalized to highlight a possible word stress position. The example of an annotation task is provided in Fig. 1. Each task was solved by three annotators. As the task is not complicated, we decide to accept only those tasks where all three annotators would agree. Finally, we obtained three sets of trigrams for the Russian, Ukrainian and Belarusian languages of approximately the following sizes 20K, 10K, 3K correspondingly. The sizes of the resulting datasets are almost proportional to the initial corpora from the Universal Dependencies treebanks.

Due to the high quality of the Universal Dependencies treebanks and the languages being not confused, there are little intersections between the datasets, i.e., only around 50 words are shared between Ukranian and Belarusian datasets and between Russian and Ukranian and Belarusian datasets. The intersection between the Ukrainian and Russian datasets amounts around 200 words.

The structure of the dataset is straightforward: each entry consists of a word trigram and a number, which indicates the position of the word stress in the central word[3].

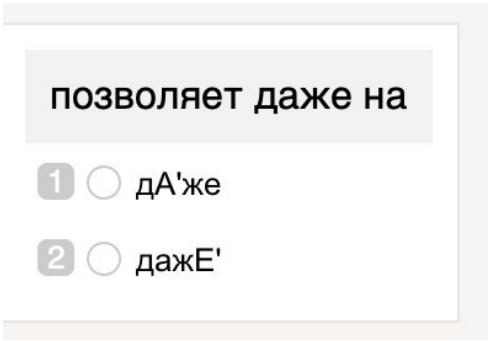

Figure 1: A screenshot of the word stress detection task from Yandex.Toloka crowdsourcing platform

3 Preprocessing

We followed a basic preprocessing strategy for all the datasets. First, we tokenize all the texts into words. Next, to take the previous and next word into account we define left and right contexts of the word as the last three characters of the previous word and last three characters of the next word. The word stresses (if any) are removed from context characters. If the previous / next word has less than three letters, we concatenate it with the current word (for example, "te_oblaká" [that-Pl.Nom cloud-Pl.Nom]). This definition of context is used since East Slavic endings are typically two-four letters long and derivational morphemes are usually located on the right periphery of the word.

Finally, each character is annotated with one of the two labels $\mathcal{L} = \{0, 1\}$: it is annotated with 0, if there is no stress, and with 1, if there should be a stress. An example of an input character string can be found in Table 1.

4 Model selection

We treat word stress detection as a sequence labeling task. Each character (or syllable) is

[1]Word stress in spoken texts database in Russian National Corpus [Baza dannykh aktsentologicheskoy razmetki ustnykh tekstov v sostave Natsional'nogo korpusa russkogo yazyka], http://www.ruscorpora.ru/en/search-spoken.html

[2]https://toloka.yandex.ru

[3]Datasets are avaialable at: https://github.com/MashaPo/russtressa

in	л	а	я		в	о	р	о	н	а		т	и	т
out	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Table 1: Character model input and output: each character is annotated with either 0, or 1. A trigram "белая ворона летит" ("white crow flies") is annotated. The central word remains unchanged, while its left and right contexts are reduced to the last three characters

labeled with one of the labels $\mathcal{L} = \{0, 1\}$, indicating no stress on the character (0) or a stress (1). Given a string $s = s_1, \ldots, s_n$ of characters, the task is to find the labels $Y^* = y_1^*, \ldots, y_n^*$, such that

$$Y^* = \arg\max_{Y \in \mathcal{L}^n} p(Y|s).$$

The most probable label is assigned to each character.

We compare two RNN-based models for the task of word stress detection (see Fig. 2 and Fig.3). Both models have a common input and hidden layers but differ in output layers.

The input of both models are embeddings of the characters. In both cases, we use bidirectional LSTM of 32 units as the hidden layer. Further, we describe the difference between the output layers of the two models.

4.1 Local model

The decision strategy of the local model (see Fig. 2) follows common language modeling and NER architectures (Ma and Hovy, 2016): all outputs are independent of each other. We decide, whether there should be a stress on each given symbol (or syllable) or not. To do this for each character we put an own dense layer with two units and a $softmax$ activation function, applied to the corresponding hidden state of the recurrent layer, to label each input character (or syllable) with $\mathcal{L} = \{0, 1\}$.

4.2 Global model

The decision strategy of the global model (see Fig. 3) follows common encoder-decoder architectures (Sutskever et al., 2014). We use the hidden layer to encode the input sequence into a vector representation. Then, we use a dense layer of n units as a decoder to decode the representation of the input and to generate the desired sequence of $\{0, 1\}$. In comparison to the local model, in this case, we try to find the position of the stress instead of making a series of local decisions if there should be a stress on each character or not.

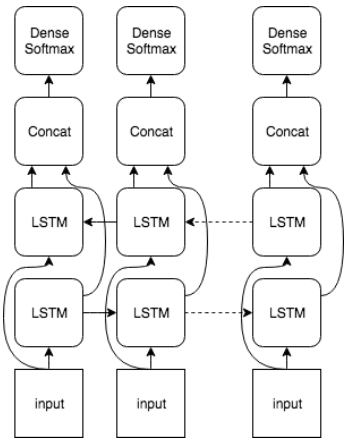

Figure 2: Local model for word stress detection

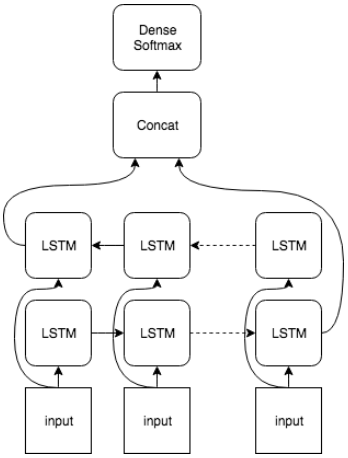

Figure 3: Global model for word stress detection

To test the approach and to compare these models, we train two models on the subcorpus of Russian National Corpus for Word stress in spoken texts, which appears to be a standard dataset for the task of word stress detection. This dataset was preprocessed according to our standard procedure, and the resulting dataset contains approximately around 1M trigrams. The results of cross-validation experiments, presented in Table 2, show that the global model outperforms significantly the local model. Hence, the global architecture is used further on in the next experiments.

We pay special attention to homographs: as one can see, in general, the quality of word stress detection is significantly lower on homographs than on regular words. However, in the majority of cases, we are still able to detect

# vowels	local	global
	all words	
2	961	983
3	940	977
4	947	976
5	960	977
6	958	973
7	924	955
8	866	923
9	809	979
avg	**952**	**979**
	homographs	
2	839	810
3	774	844
4	787	847
avg	821	819

Table 2: Accuracy scores × 1000 for two models

the word stress position for a homograph, most likely due to the understanding of the word context.

5 Experiments and results

In these series of experiments, we tried to check the following assumptions for various experiment settings:

1. monolingual setting: the presented above approach applies not only to the Russian language word stress detection but also to the other East Slavic languages

2. cross-lingual setting (1): it is possible to train a model on one language (e.g., Ukrainian) and test it on another language (e.g., Belarusian) and achieve results comparable to monolingual setting

3. cross-lingual setting (2): training on several languages (e.g., Russian and Ukrainian) will improve the results of testing on a single language (e.g., Russian) in comparison to the monolingual setting.

To conduct the experiments in these mono- and cross-lingual settings, we split the annotated datasets for Russian, Ukrainian and Belarusian randomly in the 7:3 train-test ratio and conducted 20 runs of training and testing with different random seeds. Afterward, the accuracy scores of all runs were averaged. The Table 3 presents the results of these experiments.

	test dataset		
train dataset	Belarusian	Russian	Ukrainian
Belarusian	647	326	373
Russian	495	738	516
Ukrainian	556	553	683
Ukrainian, Belarusian	769	597	701
Russian, Belarusian	740	740	563
Russian, Ukrainian	627	756	700
Russian, Ukrainian, Belarusian	772	760	698

Table 3: Accuracy scores × 1000 for different train and test dataset combinations

The Table 3 shows, that:

1. in monolingual setting, we can get high-quality results. The scores are significantly lower than the scores of the same model on the standard dataset, due to the smaller sizes of the training datasets. Nevertheless, one can see, that our approach to word stress detection applies not only to the Russian language data, but also to the data in the Belarusian and Ukrainian languages;

2. cross-lingual setting (1): the Belarusian training dataset, being the smallest one among the three datasets, is not a good source for training word stress detection models in other languages, while the Ukrainian dataset stands out as a good source for training word stress detection systems both for the Russian and Belarusian languages;

3. cross-lingual setting (2): adding one or two datasets to the other languages improves the quality. For example, around 10% of accuracy is gained by adding the Russian training dataset to the Belarusian training dataset, while testing on Belarusian.

One possible reason for the difference of Belarusian from the other two languages can be the following. After the orthography reform in 1933, the cases of vowel reduction in the unstressed position (common phonetic feature for East Slavic languages) have been represented orthographically in the Belarusian language. However, the size of the Belarusian dataset (it is much smaller than the other two) may affect the quality as well.

6 Related Work

6.1 Char-RNN models

Several research groups have shown that character-level models are an efficient way to deal with unseen words in various NLP tasks, such as text classification (Joulin et al., 2017), named entity recognition (Ma and Hovy, 2016), POS-tagging (Santos and Zadrozny, 2014; Cotterell and Heigold, 2017), dependency parsing (Alberti et al., 2017) or machine translation (Chung et al.). The character-level model is a model which either treats the text as a sequence of characters without any tokenization or incorporates character-level information into word-level information. Character-level models can capture morphological patterns, such as prefixes and suffixes so that the model can define the POS-tag or NE class of an unknown word.

6.2 Word stress detection in East Slavic languages

Only a few authors touch upon the problem of automated word stress detection in Russian. Among them, one research project, in particular, is worth mentioning (Hall and Sproat, 2013). The authors restricted the task of stress detection to find the correct order within an array of stress assumptions where valid stress patterns were closer to the top of the list than the invalid ones. Then, the first stress assumption in the rearranged list was considered to be correct. The authors used the Maximum Entropy Ranking method to address this problem (Collins and Koo, 2005) and took character bi- and trigram, suffixes and prefixes of ranked words as features as well as suffixes and prefixes represented in an "abstract" form where most of the vowels and consonants were replaced with their phonetic class labels. The study features the results obtained using the corpus of Russian wordforms generated based on Zaliznyak's Dictionary (approx. 2m wordforms). Testing the model on a randomly split train and test samples showed the accuracy of 0.987. According to the authors, they observed such a high accuracy because splitting the sample randomly during testing helped the algorithm benefit from the lexical information, i.e., different wordforms of the same lexical item often share the same stress position. The authors then tried to solve a more complicated problem and tested their solution on a small number of wordforms for which the paradigms were not included in the training sample. As a result, the accuracy of 0.839 was achieved. The evaluation technique that the authors propose is quite far from a real-life application which is the main disadvantage of their study. Usually, the solutions in the field of automated stress detection are applied to real texts where the frequency distribution of wordforms differs drastically from the one in a bag of words obtained from "unfolding" of all the items in a dictionary.

Also, another study (Reynolds and Tyers, 2015) describes the rule-based method of automated stress detection without the help of machine learning. The authors proposed a system of finite-state automata imitating the rules of Russian stress accentuation and formal grammar that partially solved stress ambiguity by applying syntactical restrictions. Thus, using all the above-mentioned solutions together with wordform frequency information, the authors achieved the accuracy of 0.962 on a relatively small hand-tagged Russian corpus (7689 tokens) that was not found to be generally available. We can treat the proposed method as a baseline for the automated word stress detection problem in Russian.

The global model, which is shown to be the best RNN-based architecture for this setting of the task, was first presented in (Ponomareva et al., 2017), where a simple bidirectional RNN with LSTM nodes was used to achieve the accuracy of 90% or higher. The authors experiment with two training datasets and show that using the data from an annotated corpus is much more efficient than using a dictionary since it allows to consider word frequencies and

the morphological context of the word. We extend the approach of (Ponomareva et al., 2017) by training on new datasets from additional languages and conducting cross-lingual experiments.

6.3 Cross-lingual analysis

Cross-lingual analysis has received some attention in the NLP community, especially when applied in neural systems. Among a few research directions of cross-lingual analysis are multilingual word embeddings (Ammar et al., 2016; Hermann and Blunsom, 2013) and dialect identification systems (Malmasi et al., 2016; Al-Badrashiny et al., 2015). Traditional NLP tasks such as POS-tagging (Cotterell and Heigold, 2017), morphological reinflection (Kann et al., 2017) and dependency parsing (Guo et al., 2015) benefit from cross-lingual training too. Although the above-mentioned tasks are quite diverse, the undergirding philosophical motivation is similar: to approach a task on a low-resource language by using additional training data in a high-resource language or training a model on a high-resource language and fine-tune this model on a low-resource language with a probably lower learning rate.

7 Conclusion

In this project, we present a neural approach for word stress detection. We test the approach in several settings: first, we compare several neural architectures on a standard dataset for the Russian language and use the results of this experiment to select the architecture that provides the highest accuracy score. Next, we annotated the Universal Dependencies corpora for the Russian, Ukrainian and Belarusian languages with word stress using Yandex. Toloka crowdsourcing platform. The experiments conducted on these datasets consist of two parts: a) in the monolingual setting we train and test the model for word stress detection on the data sets separately; b) in the cross-lingual setting: we train the model on various combinations of the datasets and test on all three data sets. These experiments show that:

1. the proposed method for word stress detection is applicable or the Russian, Ukrainian and Belarusian languages;
2. using an additional language for training most likely improves the quality of the results.

Future work should focus on both annotating new datasets for other languages that possess word stress phenomena and further development of cross-lingual neural models based on other sequence processing architectures, such as transformers.

References

Mohamed Al-Badrashiny, Heba Elfardy, and Mona Diab. 2015. Aida2: A hybrid approach for token and sentence level dialect identification in arabic. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, pages 42–51.

Chris Alberti, Daniel Andor, Ivan Bogatyy, Michael Collins, Dan Gillick, Lingpeng Kong, Terry Koo, Ji Ma, Mark Omernick, Slav Petrov, et al. 2017. Syntaxnet models for the conll 2017 shared task. arXiv preprint arXiv:1703.04929.

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. 2016. Massively multilingual word embeddings. arXiv preprint arXiv:1602.01925.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. Computational Linguistics, 31:25–70.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual, character-level neural morphological tagging. arXiv preprint arXiv:1708.09157.

Elena B. Grishina. 2003. Spoken russian in russian national corpus. Russian National Corpus, 2005:94–110.

Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), volume 1, pages 1234–1244.

Keith Hall and Richard Sproat. 2013. Russian stress prediction using maximum entropy ranking. In Proceedings of the 2013 Conference on

Empirical Methods in Natural Language Processing, pages 879–883, Seattle, Washington, USA. Association for Computational Linguistics.

Karl Moritz Hermann and Phil Blunsom. 2013. Multilingual distributed representations without word alignment. arXiv preprint arXiv:1312.6173.

Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. 2017. Bag of tricks for efficient text classification. EACL 2017, page 427.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. One-shot neural cross-lingual transfer for paradigm completion. arXiv preprint arXiv:1704.00052.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and arabic dialect identification: A report on the third dsl shared task. In Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3), pages 1–14.

Maria Ponomareva, Kirill Milintsevich, Ekaterina Chernyak, and Anatoly Starostin. 2017. Automated word stress detection in russian. In Proceedings of the First Workshop on Subword and Character Level Models in NLP, pages 31–35.

Robert Reynolds and Francis Tyers. 2015. Automatic word stress annotation of russian unrestricted text. In Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015), pages 173–180, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1818–1826.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pages 3104–3112.