Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2017"

Moscow, May 31—June 3, 2017

RESEARCH OF A DEEP LEARNING NEURAL NETWORK EFFECTIVENESS FOR A MORPHOLOGICAL PARSER OF RUSSIAN LANGUAGE

Sboev A. G. (sag111@mail.ru)[1,2,3],
Gudovskikh D. V. (dvgudovskikh@gmail.com)[1],
Ivanov I. (honala@yandex.ru)[3],
Moloshnikov I. A. (ivan-rus@yandex.ru)[1],
Rybka R. B. (rybkarb@gmail.com)[1],
Voronina I. (irina.voronina@gmail.com)[4]

[1]National Research Center «Kurchatov Institute», Moscow, Russia;
[2]National Research Nuclear University «MEPhI», Moscow, Russia;
[3]Moscow Technological University (MIREA), Moscow, Russia;
[4]Voronezh State University, Voronezh, Russian Federation

In this study we present the method of morphological tagging on base of a deep learning neural network. The method includes two levels of an input sentence processing: individual characters level and word level. The comparison with other morphological analyzers was carried out with SynTagRus dataset in its original format of morphological characters, and its versions in Universal Dependencies formats 1.3 and 1.4. Achieved accuracies of Part-of-speech tagging: 98.34%, 98.49%, 97.60% (accordingly to each dataset). Results are a bit higher than the Google Syntaxnet accuracies and higher than the accuracies of the systems based only on Bidirectional Long short-term memory models. At the MorphoRuEval competition the method gained the third place.

Keywords: artificial neural networks, natural language processing, morphological parsing, PoS-tagging

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ НЕЙРОННЫХ СЕТЕЙ ГЛУБОКОГО ОБУЧЕНИЯ В ЗАДАЧЕ МОРФОЛОГИЧЕСКОГО РАЗБОРА РУССКОГО ЯЗЫКА

Сбоев А. Г. (sag111@mail.ru)[1,2,3],
Гудовских Д. В. (dvgudovskikh@gmail.com)[1],
Иванов И. (honala@yandex.ru)[3],
Молошников И. А. (ivan-rus@yandex.ru)[1],
Рыбка Р. Б. (rybkarb@gmail.com)[1],
Воронина И. (irina.voronina@gmail.com)[4]

[1]НИЦ«Курчатовский Институт», Москва, Россия;
[2]НИЯУ «МИФИ», Москва, Россия;
[3]Московский Технологический Университет «МИРЭА», Москва, Россия;
[4]Воронежский Государственный Университет, Воронеж, Россия

1. Introduction

Nowadays there is a tendency to apply deep learning neural networks for a “sequence to sequence” transformation of data to solve such classical tasks, as Part-of-speech tagging (PoS), named entity recognition (NER), chunking and others. But so far accuracies of these tasks are higher for methods based on vocabularies and traditional machine learning algorithms: CRF, HMM, SVM [Gareev R., Tkachenko M., Solovyev V. et al]. These methods are based on a consistent representation of each word from a sentence as a set of binary encoded categorical features. The feature set of a word includes the word form ID from dictionary, IDs of its neighbors in the window, and a set of additional features of these words, such as: the first several characters and the last several characters of the word, the presence of capital letters, etc.

We develop a method based on deep learning neural networks for the following morphological analysis tasks:

1. PoS tagging,
2. features tagging—lexical and grammatical properties determination (except PoS).

Our method is based on a two-level representation of a sentence by individual characters level (see Section 2.1.1) and level of words (Section 2.1.2), inspired by works [Nogueira dos Santos C., Zadrozny B.], [Zhiheng H., Wei X., Kai Y.], [Plank B., Søgaard A., Goldberg Y.].

An information about words lengths, prefixes, terminations is important for some tasks such as PoS and total morphological tagging. It allows to use the additional word characters information more efficiently.

As the dataset we used the SynTagRus corpora in the original format of morphological features and its representations in the forms of Universal dependencies v1.3 and v1.4 (Section 3.2). Section 3 describes the results of comparisons of the proposed approach with other methods. At the MorphoRuEval competition the described method gained the third place under the name Sagteam on the scoreboard. In Section 4 we discuss the results obtained, as well as directions for further research on the development of the proposed method.

2. Materials and methods

Further we use the following terminology. The set of morphological categories includes part of speech (PoS), gender, number, case, and others. Each morphological category includes a set of features, for example in case of PoS category these are noun, verb, adjective, etc. A full tag is the unambiguous set of morphological features of appropriate categories for a word.

2.1. Two-level deep learning neural network model

We use two different models for the full morphological tagging: the first model for PoS-tagging and the second to predict the rest of morphological features (features tagging). These models have similar topologies and training methods. In frame of PoS-tagging task each part of speech is a separate class, classes are encoded in the one-hot manner. In frame of the features tagging task, output classes consist of all the unique combinations of lexical and grammatical properties (except PoS) that exist in the train set, one-hot encoded. Such an approach allows to decrease the computational complexity of the model. However, there might emerge combinations not presented in the training set, and such examples could be classified incorrectly.

The learning of the PoS model starts from training the first level (level 1 on figure 1) using character representation of every word of the train dataset. After that, the second level (level 2 on figure 2) is trained sentence by sentence. During the level 2 training, the first level weights are additionally tuned.

The training of features model (figure 2) is performed in a similar way, exceptthatthe input data includes PoS labels of every word predicted by the PoS-tagging model. The probabilities of PoS labels from PoS model are concatenated with hidden vector of level 1 (the Word 1 PoS, Word 2 PoS, Word k PoS on figure 2). The above sequential training scheme allows to re-use the symbol encoder (“hidden layers” on figures 1, 2) for other models. We implement the proposed model in Python language with the help of the Keras framework [Keras library].

Below is a detailed look at each level of the proposed topology.

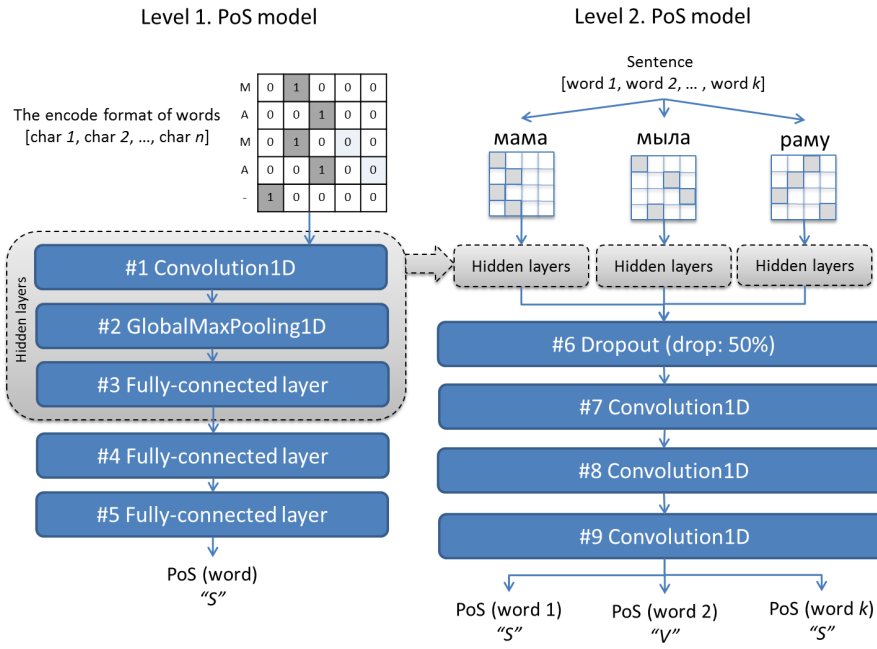

Fig. 1. The model for PoS tagging

2.1.1. The first level of the model—the individual characters representation

Words, represented as sequences of one-hot encoded characters, are the input samples of the first level. We use fixed word length, short words are extended with special "null" labels from the beginning of word. The dimensionality of an input sample is L*T, where L is the maximum word length in the training set and T is the number of unique characters in the training set + 2 ("out of vocabulary" label and "null" label). The desired class for each word is a PoS label for PoS-tagging task and a features label for features-tagging task. After training the 1st level of the network gives the vector of probabilities for desired classes. The training set consists of all words as they are in the corpus, not only unique samples.

Configuration of layers on the 1st level is identical for PoS ("Level 1. PoS model" on fig. 1) and full tag models ("Level 1. Features model" on fig. 2):

- #1 Convolution1D—convolution layer, its window goes through the word characters. Window size equals to 5 without padding on the borders of the input matrix, neuron number is 1024, activation function is ReLU [Memisevic, R., & Krueger, D.];
- #2 GlobalMaxPooling—MaxPooling over the whole word;
- #3 and #4 the fully-connected layers contain 256 neurons with activation function ReLU. The #4 layer activation values are used in level 2, described further;
- #5 Fully-connected layer, size of which is equal to the number of PoS-classes and the activation function is softmax.

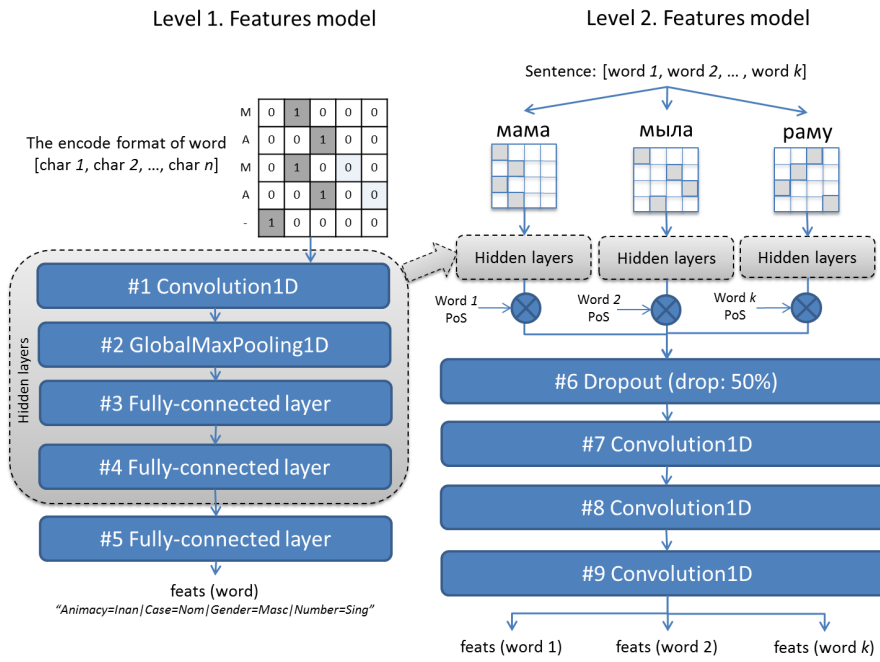

Fig. 2. The model for features tagging

2.1.2. The second level of the model—whole sentence representation

The second level input data is a whole sentence, each word of which is encoded by the activity values of a certain layer of level 1 (#3 in case of PoS-tagging and #4 in case of features tagging) in response to that word. We use fixed sentence length—the maximum length in the training set. Short sentences are extended from the beginning with "null words" consisting of "null" label characters. Such "null words" belong to special null class. The level 2 predicts labels for all words of a sentence at once.

Configuration of layers on the 2nd level (identical for PoS and full tag models):

- Layers #7 and #8 are Convolution layers, with window going through the words of the sentence. Window size equals 3 with padding on the borders of the input matrix, neurons number is 256, activation function is ReLU. A zero vector is added to the end and to the beginning of the sentence ("same" border mode in Keras).
- #9 Model output is a convolution layer, its window goes through the words, window size is 3,the neuron number is the PoS-classes number + 1 (for the zero padding),the activation function is softmax.

2.1.3. Learning configuration of two-level deep learning neural network model

We set the maximum of 300 epochs for training with early stopping: if the mean square error (MSE) stays the same or rises on a validation dataset during several consecutive epochs (15 on level 1 and 10 on level 2), training stops and neural network weights are set to the state with the minimum validation loss during training in case of PoS-tagging task, or remain at the state of the last epoch in case of the features task.

The MSE loss function is calculated for each word in the dataset on the first level training and for each padded sentence on the second level training. The optimizer is Adamax [Kingma, D., & Ba, J.]. Batch normalization function is used on the first level for activity normalization between GlobalMaxPooling and #3 layers, as well as between #4 and #5 layers. Batch size was 1024 on the 1st level and 32 on the 2nd level.

2.2. Other models for comparison

The set of well-known models were compared with the approach proposed in this paper: SVM, its extended version using Yandex.Mystem, Syntaxnet (PoS-tagging part).

2.2.1. SVM-based Approach

In this case a word is represented as the vector of word forms indices, which includes the indices of: n words to the left, the base word, k words to the right. These indices are defined on base of the learning sample dictionary. There are two rules: if the word is not in the dictionary, the ID of this word equals to 1; if in some places of window there are no words, the indices of 0 values fill these places. The ensemble of linear SVM was used, learned on base of one-vs-all strategy. The number of these classifiers equals to the number of morphological features to be defined.

2.2.2. Extension of the SVM approach

The main characteristic of this approach [Rybka, R., Sboev, A., Moloshnikov, I., Gudovskikh, D.] is to add the results of preliminary MYSTEM tagging to feature vector for the final parsing. For this purpose the MYSTEM results are transformed to tagging format of SynTagRus by the specially created converter. The list of features contains:

- All word forms from the window W;
- Tags for words of W that have been analyzed on previous steps;
- Classes of ambiguities for all words from W (+ their bigrams and trigrams). Class of ambiguity is the set of all possible tags for a word. We represent it as a concatenated tags string. For example, in case of Russian equivalent of the word “These” the sentence-example class ambiguity looks like this:

adjective|nominative_case|plural_adjective|accusative_case|plural|inanimate;

- Possible full tags for each word;
- Determined morphological features for parsed words of W;
- Possible morphological features for each unparsed word from W.

The dimension of window W equals to 7, W includes 3 words from left side and 3 words from right from the analyzed words. Words are sequentially processed from right to left. The ensemble of linear SVM was used to predict individual morphological features. Thus each classifier solves a binary classification task.

The following function is used for resulting class choice:

function (X, DV, M):
Here X=$\{x_1,...,x_K\}$ is the set of all possible full tags,
and $DV = \{dv_1,..dv_M\}$, where $dv \in [-1,+1]$, is the decision value of the classifier m
of the SVM ensemble

```
estimation_list = [] # will contain a probability of each x_k for 1 ≤ k ≤ K
  for k in range(0, K): # for each possible full tag x_k, now called x
    x = X[k]
    # a full tag x_k is a vector (v_1, ..., v_M) of morphological features v_m ∈ {0,1},
    # where M is the number of morphological features in the corpus
    similar_score = 0
    different_score = 0
    for m in range(0, M): # for each morphological feature v_m ≡ x[m]
      if (x[m] == 1) and (DV[m] > 0): similar_score+ = DV[m]
      else: different_score = different_score + DV[m]
    estimation_score_m = similar_score − different_score
    estimation_list.append(estimation_score_m)
max_index = argmax(estimation_list) # getting the index of the highest element
of estimation_list
return X[max_index] # the resulting tag is the one with the highest probability
```

2.2.3. One-level deep learning model

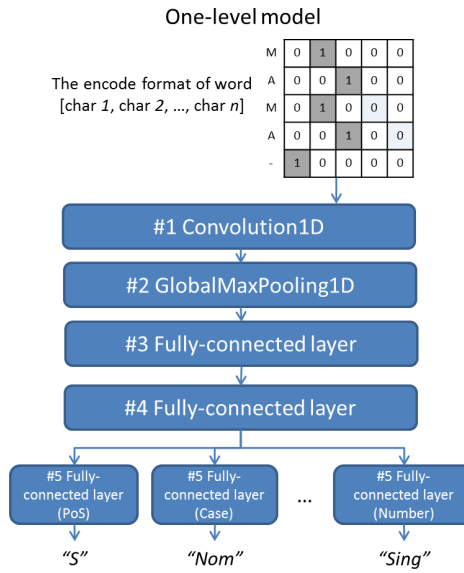

Fig. 3. The model for full morphological tag for one word

Also we added another neural network model for comparison (Fig. 3), called One-level model in Table 3. At the input this model gets character word representation without its neighborhood in the sentence. The last layer consists of several parallel equal layers, each of which corresponds to a morphological category like PoS, Case, Number etc. Layer #1 is the convolution layer, performing a character-by-character pass window size of 5. The layer contains 1,024 neurons with ReLU activation function. Keras border mode is "valid". GlobalMaxPooling #2 is max-pooling over time; #3 and #4 are fully-connected layers, each having 256 ReLU neurons; #5 layers are

fully connected layers with dimensionality equal to the number of features in each category (PoS, Case, Number etc.), with linear or softmax activation functions.

3. Evaluation

3.1. Prediction scores

Accuracy metric and weighted F1-score were chosen as comparison criteria. We present scores separately for testing datasets and for words not existing in the training dataset, further called out-of-vocabulary (OOV).

3.2. Used corpora

We used SynTagRus dataset with different formats of morphological features: original format contained in National Russian Corpus, Universal dependencies (UD) 1.3 and 1.4. In case of UD dataset format we used the original predefined splitting into training, testing, and developing sets. SynTagRus with original format was split into 3 parts manually. The datasets differ in number of sentences (Table 1), type of PoS-tags, and other morphological features (Table 2).

Table 1. Number of sentences and tokens in various datasets formats

SynTagRys datasets type	Number of sentences in			Number of tokens in		
	Train set	Test set	Dev set	Train set	Test set	Dev set
Original	47,980	5,923	5,331	695,255	86,163	77,249
UD-1.3	46,750	6,130	6,250	815,485	107,737	109,422
UD-1.4	48,171	6,130	6,250	850,689	108,100	109,694

Table 2: Number of unique PoS-tags, morphological features, and different full tags

SynTagRys datasets type	Number of PoS features	Number of morphological features	Number of uniques full tags	
			All corpus	Train set
Original	11	45	450	447
UD-1.3	15	36	436	433
UD-1.4	16	36	585	581

3.3. Experiments

Accuracy and F1-score metrics were calculated for the following models:

- Linear SVC (described in Section 2.2.1),
- the approach based on an extended linear SVC combined with Yandex.MyStem (described in Section 2.2.2), the window size equals 8,

- the proposed two-level deep learning neural network model approach (described in Section 2.1), with dropout and without dropout,
- the model involving a single level for full tag based only on character representation of a word (described in Section 2.2.3), called "one-level model" in Table 3.

The results are presented in table 3.

Table 3. Accuracy and F1-score of different models on SynTagRus datasets in different formats

Type of SyntagRus format	Model name	all				OOV			
		POS		full		POS		full	
		accuracy	F1 score	accuracy	F1 score	accuracy	F1 score	accuracy	F1 score
Original	Extended LinearSVC: window size=±3	94.10	94.05	83.90	91.24	63.22	61.90	29.70	45.80
	Extended LinearSVC: window size=±2	94.39	94.34	85.04	91.91	62.11	60.00	30.20	46.40
	Extended LinearSVC: window size=±1	95.02	94.98	85.74	92.32	63.33	61.40	29.70	45.80
	Extended LinearSVC+ Mystem (window size of 8)	95.61	96.00	81.65	89.90	95.91	96.30	79.60	88.70
	One-level model	96.63	96.64	85.58	92.23	94.72	94.74	74.76	85.56
	Proposed approach	98.24	98.23	94.12	96.97	95.14	95.20	84.40	91.60
	Proposed approach + Dropout	**98.34**	**98.33**	**94.83**	**97.35**	**95.24**	**95.25**	**85.07**	**91.93**
Universal Dependencies 1.3	Extended LinearSVC: window size=±3	94.87	94.84	82.30	90.29	69.22	67.90	13.32	23.51
	Extended LinearSVC: window size=±2	95.20	95.17	83.33	90.91	69.17	67.47	12.60	22.38
	Extended LinearSVC: window size=±1	95.46	95.41	84.04	91.33	68.85	65.85	11.91	21.28
	One-level model	96.85	96.82	85.56	92.22	94.13	94.19	59.86	74.89
	Proposed approach	98.44	98.44	93.34	96.55	95.16	95.20	71.30	83.25
	Proposed approach + Dropout	98.49	98.49	94.31	97.07	95.07	95.09	74.48	85.37
	GOOGLE	98.27	98.27	94.01	96.92	94.21	94.35	74.12	85.13
Universal Dependencies 1.4	Extended LinearSVC: window size=±3	93.98	93.91	81.59	89.86	61.08	60.12	11.73	21.00
	Extended LinearSVC: window size=±2	94.31	94.25	82.79	90.59	60.97	59.90	12.05	21.50
	Extended LinearSVC: window size=±1	94.46	94.38	83.46	90.98	60.54	59.00	10.46	18.90
	One-level model	95.60	95.54	84.50	91.60	85.71	85.47	56.63	72.31
	Proposed approach	97.51	97.49	92.79	96.26	88.63	88.53	69.32	81.89
	Proposed approach + Dropout	**97.60**	**97.58**	**93.44**	**96.61**	88.34	88.06	**70.22**	**82.50**

Table 3 shows the following:

1) LinearSVC window size increasing does not give better accuracy.
2) The approach based on Extended LinearSVC and MyStem gives better accuracy than LinearSVC in case of out-of-vocabulary words prediction.
3) Neural network model with the one-level topology ("One-level model") gives accuracy similar to LinearSVC ones, but shows worse results in out-of-vocabulary words parsing.
4) The proposed approach shows accuracy a bit higher than the Google parser.

3.4. MorphoRuEval on Dialog 2017

As part of the competition, we used a modified version of the two-level model. We add the Batch normalization layers between layers #2 and #3 and between layers #4 and #5 in level 1. The model is trained on the GICRYA corpus, provided by the organizers. The corpus was divided into a training (90%) and validation set (10%). Testing was performed on three datasets (not disclosed to the competition participants): news, posts in a social network ("social media" in Table 4) and fiction literature. For each dataset two tasks were graded, full tagging and lemmatization, and two accuracy measures were evaluated for each task, the ratio of words correctly classified and the ratio of sentences completely correct.

Table 4. The results of the model for each measure compared to a few leaders

Dataset	Task	Accuracy measure	First place (%)	Second place (%)	This study (%)	Fourth place (%)	Fifth place (%)
News	Full tagging	Accuracy on words	93,71	93,99	93,35	93,83	90,52
		Accuracy on sentences	64,80	63,13	55,03	61,45	44,41
	Lemmatization	Accuracy on word forms		92,96	81,6	93,01	
		Accuracy on sentences		56,42	17,04	54,19	
Social Media	Full tagging	words	92,29	92,39	92,42	91,49	89,55
		sentences	65,85	64,08	63,56	61,44	51,41
	Lemmatization	word forms		91,69	82,8	90,97	
		sentences		61,09	35,92	60,21	
Fiction literature	Full tagging	words	94,16	92,87	92,16	92,4	90,13
		sentences	65,23	60,91	56,6	60,15	48,48
	Lemmatization	word forms		92,01	77,78	91,46	
		sentences		57,11	22,08	55,08	
Mean	Full tagging	words	93,39	93,08	92,64	92,57	90,07
		sentences	65,29	62,71	58,4	61,01	48,1
	Lemmatization	word forms		92,22	80,73	91,81	
		sentences		58,21	25,01	56,49	

4. Conclusion

The presented results demonstrate the great potential of complicated deep learning models compared to traditional SVM ones. The approach on base of MYSTEM is more effective in case of words not presented in the training set. This fact is expected since the latter approach is based on common dictionaries and linguistic rules without tuning to any corpus. As a result it loses in comparison to deep learning models in specific cases. For practical needs it would be useful to unite these approaches in a common morphological parser to increase the universality and the accuracy of parsing.

Acknowledgments

The reported study was funded by RFBR according to the research project № 16-37-00214

References

1. *Gareev R., Tkachenko M., Solovyev V., Simanovsky A., Ivanov V.* (2013), Introducing baselines for Russian named entity recognition. Volume 7816 of the series Lecture Notes in Computer Science, pp. 329–342.
2. Keras library [Online]. Available: http://keras.io
3. *Kingma, D., & Ba, J.* (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
4. *Memisevic, R., & Krueger, D.* (2014). Zero-bias autoencoders and the benefits of co-adapting features. stat, 1050, 13.
5. *Nogueira dos Santos C., Zadrozny B.* (2014), Learning Character-level Representations for Part-of-Speech Tagging, Proceedings of the 31st International Conference on Machine Learning (ICML) — Volume 32, Beijing, China, pp. II-1818 — II-1826.
6. *Plank B., Søgaard A., Goldberg Y.* (2016), Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss, Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, pp. 412–418.
7. Russian national corpus (2016) [Online]. Available: http://ruscorpora.ru/en
8. *Rybka, R., Sboev, A., Moloshnikov, I., Gudovskikh, D.* (2015, November). Morpho-syntactic parsing based on neural networks and corpus data. In Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015 (pp. 89–95). IEEE.
9. *Zhiheng H., Wei X., Kai Y.* (2015), Bidirectional LSTM-CRF Models for Sequence Tagging, availiable at: arXiv:1508.01991.