Neural network for generating wordforms using lemmas and morphological features

Moloshnikov I.A. (ivan-rus@yandex.ru), Vlasov D.S. (vfked0d@gmail.com),
Gryaznov A.V. (artem.official@mail.ru),
Gribov N.A. (griboet@yandex.ru),
Sboev A.G. (sag111@mail.ru)

NRC "Kurchatov Institute", Moscow, Russia

Abstract: The article accompanies the participation in the competition for the morpho-analysis of languages with a small amount of resources. In one of the tasks, it was proposed to generate a word form for a given lemma and set of grammatical labels.For the first time a method has been proposed for the generation of word forms for low-resource languages: Evenki and Selkup. The basis of the proposed method is a neural network based on long short-term memory layers. The corpus in the format of Universal Dependencies is used as a data set. Using the proposed model for generating word forms allowed to reach 53% for Evenki and 50% for Selkup.

Key words: generating wordforms, neural network, text analysis, machine learning, deep learning in NLP, corpus linguistics, low-resource languages

Молошников И.А. (ivan-rus@yandex.ru), Власов Д.С. (vfked0d@gmail.com),
Грязнов А.В. (artem.official@mail.ru),
Грибов Н.А. (griboet@yandex.ru),
Сбоев А.Г. (sag111@mail.ru)

НИЦ "Курчатовский институт Москва, Россия

Аннотация: Статья сопровождает участие в соревновании по морфоанализу языков с малым количеством ресурсов. В одной из задач было предложено по заданной лемме и набору грамматических меток сгенерировать словоформу. В этой статье впервые предложен метод для генерации словоформ для малоресурсных языков: эвенкийский и селькупский. Основой предложенного метогда является нейронная сеть на основе long short-term memory layers. В качестве набора данных использован корпус в формате Universal Dependencies. Использование предложенной модели для генерации словоформ позволило достигнуть 53 % на эвенкийском и 50 % на селькупском языках.

Key words: генерация словоформы, нейронные сети, анализ текстов, машинное обучение, глубокое обучение естественного языка, корпусная лингвистика, малоресурсные языки

1 Introduction

With the development of technology and new types of communication between people, there is a need to process natural languages. Already today a large number of word processing tools are available in the form of dictionaries, machine translators, speech synthesis and speech recognition systems. However, such systems are only being actively developed for languages for which the necessary language and speech electronic resources are available. The majority of the languages today are still little studied. However, in recent years more and more attention has been paid to creating systems for automatic processing of low-resource languages. The concept of a low-resource language refers to natural languages which have a number of properties:

1. lack of linguists and language translators;

2. low distribution on the Internet

3. insufficient number of electronic resources, corpora, bilingual dictionaries.

The properties listed above significantly complicate the development of methods for automated processing of such languages. The article proposes an approach to solving the problem of synthesizing the word form of a language based on the available lemmas and morphological features of the desired word form. Such word form synthesizing solution would be useful for various text generation systems when one needs to build a word with a specific morphological structure.

One modern approach [7] is sequence-to-sequence, when a new sequence of characters or words is generated based on the input sequence of characters or words. This approach has been successfully applied to machine translation systems [6], generating word sequences [4]. We here propose to borrow the sequence-to-sequence approach for solving the problem of word form synthesis and word form lemmatization. The proposed method is based on a neural network model with recurrent layers. Experiments are carried out on the corpora for Evenki and Selkup languages in the format of Universal Dependencies.

2 Method

2.1 Short description

In this work, we used a model based on LSTM layers [3]. When synthesizing the wordforms of the lemma are given in the form of sequences of characters with regard to morphological features. LSTM is one of the varieties of artificial recurrent networks in which gate mechanisms are implemented. LSTM have been designed to handle large data sequences in order to solve the gradient attenuation problem when using the back propagation error method when training a network. This kind of networks shows good results in various tasks related to word processing.

2.2 Data

As part of the competition, the resource languages listed in table 1 , were proposed, but the synthesis of the word form was carried out only for those languages for which there was a sufficient amount of input data (lemmas and morphological features) for conducting experiments.

Since in the presented approach the synthesis of the word form is based on the lemmas and morphology, it is assumed that they are necessary for the construction of the model. Then the selected languages for the study are Evenki and Selkup. It is noted that these corpus contain wordforms that have the same lemmas and the same morphological features, which complicates the decision.

2.3 Data preprocessing

Since the LSTM network needs to submit vector representations of the text, the following data coding was performed:

1. for encoding lemmas in the vector, a frequency dictionary of symbols was assembled along the corporus; based on the 50 most common characters, each character of the

Table 1: Comparison of the enclosures provided

Language	**Count of wordforms**	**Count of wordforms which there are no lemmas or moprhtags**
Evenki	26926	1057
Selkup [1]	13436	0
Karelian (proper Karelian dialect [8])	68751	56445
Karelian (livvik dialect)	66534	27212
Karelian (lyudikov dialect)	16391	9530

word was encoded with a vector of 0 or 1. The so-called unitary coding - One Hot Encoding (OHE). Constant 50 was chosen empirically by running several experiments.

2. For coding morphological features, a dictionary of all morphological features was compiled, on the basis of which coding was carried out in the same way.

Thus, one example of the word form was represented as a character-by-character concatenation of OHE representations of the characters of the lemma and OHE by the representation of morphological features, which allows to take into account the lemma and morphology. When this morphological features is duplicated for each character of the word form, due to which the morphology is not forgotten when learning the network.

3 Network topology

For training, a bidirectional LSTM network was built, into which the entire coded lemma and the padding vector of morphological features were fed up to the maximum length of the word form:

1. Input 1, Input2: 2 inputs: 1 is the vector of the lemma, 2 is the vector of the morphological tag.
2. Concatinate layer: the concatenation layer of two input vectors.
3. Bidirectional LSTM: a layer with 32 neurons
4. Bidirectional LSTM: a layer with 32 neurons
5. Dense layer: with 50 neurons (since the coding was on the 50 most frequency symbols) with the softmax activation function.

The network topology is also shown in Figure 1. To obtain a vector representation of lemmas and morphological features, a batch generator was implemented, receiving input for lemmas, morphtags and dictionaries for encoding, and output for giving them matrix representations. After that, the resulting matrix representations are concocted and transmitted to the bidirectional LSTM network, at the output of which a vector is formed, which subsequently falls into a fully connected layer with the softmax activation function.

Training parameters for the model:

- batch size 32;

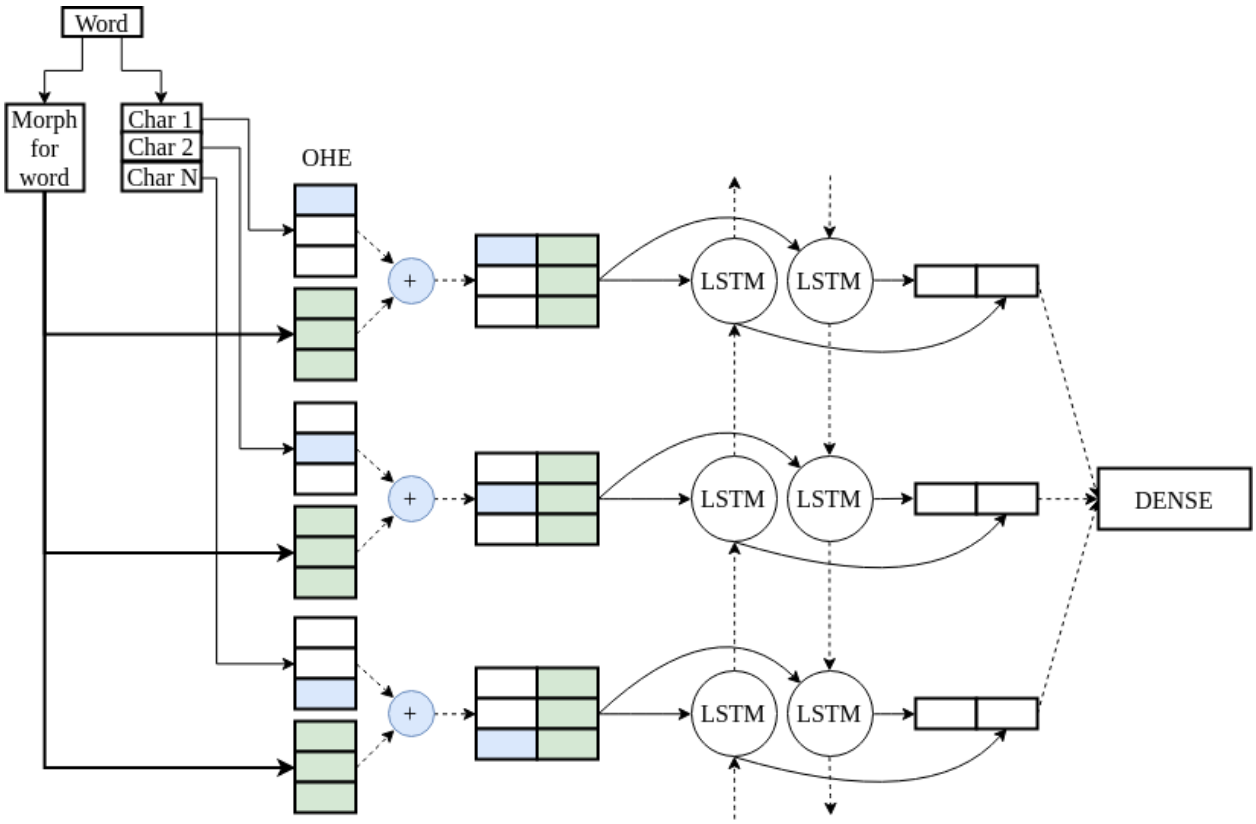

Figure 1: Network topology

- Optimizer: Adam [5];
- Optimization evaluation function - accuracy;
- The loss function is binary crossentropy;
- Early stop after 150 epochs; the maximum number of epochs is 3000.

The ratio of the separation of training and validation sets is 80 to 20.

4 Results

4.1 Metrics

To assess the accuracy of the model, the metrics proposed as part of the competition were used:

- proportion of fully correctly generated word forms
- Levenshtein average distance between the word form and the correct answer

The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

4.2 Testing the method on the task lemmatization of wordform

As data for training were taken materials provided in the framework of the competition CoNLL-2018. Word forms and morphological signs were input to the network, coding was carried out on the basis of the 35 most common characters. The network was trained in the same way as in the synthesis of the word form. The table 2 presents the results of solving the problem of lemmatization for various languages and for the given corpus of the Evenki language.

Table 2: The results of solving the problem of lemmatization

Language	**Accuracy of the lemma definition**		**Count wordforms**		
	CoNLL2018 baseline	**Our model**	**Train size, 10^3**	**Valid size,10^3**	**Test size,10^3**
Afrikaans AfriBooms	**97.6**	95.77	30.504	3.390	5.317
Latvian LVTB	92.0	**92.66**	72.737	8.082	14.637
Russian SynTagRus	95.9	**96.92**	784.368	87.153	118.691
Irish IDT	86.5	**88.89**	12.443	1.383	0.144
Galician CTG	97.0	**98.54**	78.008	8.668	32.521
English GUM	**96.2**	90.45	48.317	5.369	13.164
French Sequoia	**97.8**	92.08	46.715	5.191	10.294
Evenki	-	93.98	18.848	1.615	6.463

As a result of testing, we conclude that the use of such a method effectively performs the task of lemmatization.

4.3 Wordform synthesis

The task of generating word forms was previously set at the competition CoNLL – SIGMORPHON 2018 [2]. Given an input lemma and desired output tags, participants had to generate the correct output inflected form (a string). The results obtained with an indication of the accuracy of the declared metrics are shown in table 3.

Table 3: The results of testing the model

Language	**Proportion of fully correctly generated wordforms**	**Mean the Levenshtein distance**
Evenki	0.53	1.258
Selkup	0.50	1.162

Using the example of Evenki, we note that the network has learned how to recover lemmas well, but has not learned how to do the exact inverse transformation. Perhaps this is for the following reasons:

1. in the data there is an ambiguity between the set of the lemma and morphological features with the corresponding word form.

2. map the space of "lemma + morphological features" into the space of "word forms" is a more facilitated task than the "word forms + morphological features" in the "lemmas". This is related to the dimensions of these spaces. The number of unique "lemma + morphological features" is 10181 and the corresponding number of unique word forms is 11822, the number of unique "word forms + morphological features" is 12384, despite the fact that the number of unique "lemmas" is 5355.

5 Conclusion

For the first time, a method for generating word forms for low-resource languages is proposed: Evenki and Selkup. The method is based on the sequence to sequence approach and is implemented using layers of a recurrent neural network. The method was tested on the task of lemmatization, the results are comparable with the accuracy of the baseline of the CONLL 2018 competition. Using the proposed method to generate word forms allowed to reach 53% for Evenki and 50% for Selkup. The result can be used as a base accuracy.

6 Acknowledgments

The reported study was funded by RFBR according to the research project №18-37-00331 "mol_a".

References

[1] Wagner-Nagy Beáta Brykina Maria, Orlova Svetlana. Inel selkup corpus. pages 172–177, 2018.

[2] Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. The CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection. *arXiv e-prints*, page arXiv:1810.07125, Oct 2018.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[4] Glorianna Jagfeld, Sabrina Jenne, and Ngoc Thang Vu. Sequence-to-Sequence Models for Data-to-Text Natural Language Generation: Word- vs. Character-based Processing and Output Diversity. *arXiv e-prints*, page arXiv:1810.04864, Oct 2018.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Graham Neubig. Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. *arXiv e-prints*, page arXiv:1703.01619, Mar 2017.

[7] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q.

Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[8] Krizhanovsky A.A. Pellinen N.A. Rodionova A.P. Zayceva N.G., Krizhanovskaya N.B. Open corpora vepsian and karelian languages (ВепКар):preliminary selection of materials and the vocabulary part of the system, proceedings of the international conference "corpus linguistics - 2017.