# The usage of the word order change algorithm for the identification of emotionally charged texts in Russian

**V Barakhnin[1,2*], O Kozhemyakina[1], I Pastushkov[1]**

[1]*Institute of Computational Technologies of SB RAS, Lavrentiev av., 6, 630090, Novosibirsk, Russia*
[2]*Novosibirsk State University, Pirogov str.,1, 630090, Novosibirsk, Russia*

**Corresponding author's e-mail: bar@ict.nsc.ru*

**Abstract**

This paper describes the algorithm for the so-called "straightening" of word order, which is used for preprocessing texts with computer analysis methods. "The straightening" improves the quality of the corresponding algorithms. In addition, a sufficiently large Damerau–Levenstein distance between the converted and the source text can be used as a sign of emotional coloring of the text, which is used, for example, to give it the character of political manipulation.

*Keywords:* automated analysis, "straightening" of word order, Damerau–Levenstein distance, emotional coloring of the texts

## 1 Introduction

One of the most common figures of speech is the inversion – "violation" of the «natural» word order". However, when we use the algorithms for computer analysis of texts, the sentences with inversion can cause the errors. This applies to both "direct" algorithms, since their creators usually proceed from the "natural" word order, and machine learning algorithms, since the main part of sentences in the average corpus of texts that algorithms are trained on has a "natural" word order.

This paper describes the algorithm for the so-called "straightening" of word order, originally designed to adapt the word order in poetic texts for the purpose to use the popular concept of *word2vec* for their classification [1], as well as machine learning methods on large-volume text corpuses, such as *Syntagrus* [2]. Of course, this "straightening" improves the quality of computer analysis algorithms for prose texts.

In addition, the word order is important for determination of the emotional color of a text. Our hypothesis is that the quantitative metric of word order bias can be used as a sign of emotional coloring of the text, applied, for example, to give it the character of political manipulation.

The idea of the method is as follows:

1. To bring the text to the "natural" (grammatical) word order which is inherent to scientific and serious journalistic texts.
2. To encode each sentence with a sequence of characters, where each character corresponds to a word.
3. To encode the original text in the same way.
4. To calculate the Damerau–Lowenstein distance for each pair of character sequences [3, 4].
5. Calculate the median value for the entire text.

To calculate the degree of discrepancy in word order, the Damerau-Levenstein distance was used, and not the classical Levenstein distance, since the original algorithm does not contain a transposition operation, i.e., a permutation of characters, and, based on the implementation of point 1, which will be described later, the source text and received text contain the same lexemes, and thus the operations of insertion, replacement and deletion will simply not appear in the calculation.

## 2 Model for changing word order

The word order depends on syntactic constructions, which are further referred to as syntactic groups – by analogy with the classical work of N. Chomsky [5]. We propose a modification of the general idea of methods for selecting syntactic groups based on the correction of the responses given by the classifier using machine learning, with a probability below a given threshold using the so- called statistical classifier.

We have considered the following basic models:

1. The multi-layer perceptron.
2. The XGBoost − the most effective implementation of gradient descent.
3. The logistic regression.

To get the best result on a small corpus it was suggested to use the ensembling of these methods together with data preprocessing:

1. The selection which is based on the probability threshold value (in our case, the threshold value was taken to be 0.86).
2. The stacking – the usage of a meta-algorithm over the results of classifiers (in this problem – the logistic regression over the previously specified algorithms).

The calculation results are presented in Table 1.

The table shows that the best results are obtained by the multi-layer perceptron with data preprocessing by selecting a probability threshold value with a statistical classifier.

TABLE 1 Chunking methods comparison

| Combination method | Second classifier | Avg | Max | Min |
|---|---|---|---|---|
| Stacking with Statistic classifier | XGBoost | 0.9 | 0.91 | 0.89 |
| | MLP | 0.91 | 0.92 | 0.87 |
| | Logistic regression | 0.88 | 0.89 | 0.87 |
| Threshold-based combination with statistic classifier | XGBoost | 0.91 | 0.93 | 0.9 |
| | MLP | 0.92 | 0.94 | 0.9 |
| | Logistic regression | 0.88 | 0.89 | 0.87 |

## 3 The algorithm of the changes of word order

The "word bag" approach works well for the text classification tasks. This approach assumes that the presence or absence of words matters more than the sequence of words. However, there are problems when we use it: when recognizing entities, identifying parts of speech, and so on, the sequence of words is no less important. The conditional random fields (CRF) come to the rescue here, because they use sequences of words, not just individual words. Below is a formula for CRF, where *y* is the hidden state (for example, part of speech), and *x* is the observed variable (in this example, it is an entity or other words around it):

$$p(\mathrm{y}|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} \exp\{\sum_{k=1}^{K} \eta_k f_k(y_t, y_{t-1}, x_t)\}.$$

The CRF model is an undirected graph model that takes into account the words that occur not only before an entity, but also after it. The parser divides text into syntactically related parts of a sentence. The next step is to pre-process training and testing data to extract the attributes that express the characteristics of words (elements) in the data.

We trained the classifier on a standard set of features for selecting syntactic groups. In our case, it all comes down to the problem of multi-class classification, where the objects are the words in sentences, and the target variables are the BIO-tags. The following attributes were selected:

- the morphology of the word-object;
- the morphology of words to the left of the object word;
- the morphology of words to the right of the object word.

The context ±2 was selected, i.e. the morphology of two tokens before and after was considered from the specified word.

It was suggested to use the Python shell for CRFSuite for the task of highlighting of the syntactic groups. The training corpus was automatically converted according to the templates to the CRFsuite format, which in this case is used to get the attributes specified by the template, and then the logistic regression is applied to the feature matrix. This method showed 87 % accuracy in the test sample, and the group found as a whole is considered correctly found, not just individual tags which are found correctly.

The algorithm that allows for the so-called "straightening" of word order is based on the algorithm for text selection of syntactic groups and on the usage of the results of its operation as the input of a recurrent neural network, the experiments were conducted using a neural network with long short-term memory( LSTM) [6] and the CRFSuite utility.

The algorithm demands next preparation steps:

1. Syntax chunker that we describe above training on modified corpus of *SynTagRus* which contains also a different variants of word order, equivalent to unordered *n*-grams used in [7] so the word order entropy is taken into account by synthetic data addition.
2. By the *pymorphy2* [8] module usage, the morphological analysis is performing and corresponds each word with its syntax features and sends it to input of recurrent neural network.

The steps of the algorithm:

1. A recurrent neural network works as a classifier that matches each group and its context (a window in 2 groups before and after) with an offset from -2 to 2, which corresponds to where the word needs to be shifted. A value of 0 means that the word is in its proper place.
2. A Python script takes probabilities into account and dynamically rearranges the syntactic groups.
3. A neural network of similar topology also classifies the words within groups by displacement class, based on their morphological properties.
4. A Python script, taking probabilities into account, dynamically rearranges words by analogy with syntactic groups.

As two examples of texts for testing the algorithm from the site https://tengrinews.kz/ two rexts were chosen which are deliberately colored emotionally, since it operates on the resonant topic of AIDS, and deliberately unemotional about the assignment of titles to the highest leaders of the Kazakhstan national security committee. The calculations using the algorithm showed that the Damerau-Levenstein distance for the first example is 6, for the second example is 0.

## 4 Conclusions

This paper describes the algorithm for the so-called "straightening" of word order, which is used for preprocessing texts with computer analysis methods. "The straightening" improves the quality of the corresponding algorithms. In addition, a sufficiently large Damerau–Levenstein distance between the converted and the source text can be used as a sign of emotional coloring of the text, which is used, for example, to give it the character of political manipulation.

## Acknowledgments

The work was funded by grant No BR05236839 of the Ministry of Education and Science of the Republic of Kazakhstan, by the Russian Fund of Basic Research, project No 19-31-27001 and within the framework of the state task theme No AAAA-A17-117120670141-7 (No 0316-2018-0009).

## References

[1] Mikolov T, Chen K, Corrado G, Dean J 2013 Distributed representations of words and phrases and their compositionality *Advances in neural information processing systems* 3111-19
[2] Dyachenko P, Iomlin J, Lazurskiy A, Mityushin L, Podlesskaya O, Sizov V 2015 The current state of the deeply annotated corpus of texts of the Russian language (SynTagRus)
[3] Дяченко П, Иомдин Л, Лазурский А, Митюшин Л, Подлесская О, Сизов В 2015 Современное состояние глубоко аннотированного корпуса текстов русского языка (СинТагРус*) Proceedings of the Institute of Russian language. Named after V. V. Vinogradov (6) 272-300 (in Russian)*
[4] Damerau F 1964 A technique for computer detection and correction of spelling errors *Communications of the ACM* **7**(3) 171-6
[5] Levenstein V 1965 The binary codes with the correction of drops, inserts and substitutions of characters *Reports of the Academy of Sciences of the USSR Доклады Академии наук СССР* **163**(4) 845-8 (*in Russian*)
[6] Chomsky N 2014 *The minimalist program* MIT press
[7] Hochreiter S, Schmidhuber J 1997 Long short-term memory *Neural Computation journal* **9**(8) 1735-80
[8] Barakhnin V, Kozhemyakina O, Pastushkov I 2017 Automated determination of the type of genre and stylistic coloring of Russian texts *ITM Web of Conferences* https://doi.org/10.1051/
[9] itmconf/20171002001
[10] Korobov M 2015 Morphological analyzer and generator for Russian and Ukrainian languages *International Conference on Analysis of Images, Social Networks and Texts* 320-32