Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2016"

Moscow, June 1–4, 2016

NLP PIPELINE FOR RUSSIAN: AN EASY-TO-USE WEB APPLICATION FOR MORPHOLOGICAL AND SYNTACTIC ANNOTATION

Droganova K. A. (kira.droganova@gmail.com),
Medyankin N. S. (nikita.medyankin@gmail.com)

School of Linguistics, Faculty of Humanities, Higher School of Economics, Moscow, Russia

The aim of this paper is to present an easy-to-use web application specifically developed for annotating Russian texts with morphological and syntactic information. The application is built upon the pipeline that utilizes the same basic ideas as in the experiments conducted by Serge Sharoff on Syntagrus. However, different tools and models are used. We have put an extensive effort into development of our own rule-based segmentation module, which showed 99.5% accuracy. Tokenization, lemmatization, and morphological tagging are conducted via Mystem. Morphological information is disambiguated using TreeTagger with parameter model trained on disambiguated part of Russian National Corpus. Accuracy of morphological annotation for full tag set measured in a strict sense (i.e., one missing or misplaced tag for a token is a miss, full match tag-by-tag is a hit) is 85.5%. Precision and recall of morphological annotation for full tag set measured in classical sense are 92.4% and 91.6% respectively. Syntactic annotation is obtained via MaltParser using a specifically trained model with the quality of 83.7% by LAS and 89.6% by UAS. The use of the application under consideration does not require any specific technical knowledge or software, therefore making automatic morphological and syntactical annotation of texts easily available for any person with the Internet access. Furthermore, since it uses the same tagset as Russian National Corpus, it provides the means for obtaining morphologically and syntactically pre-annotated corpus of Russian texts compatible with RNC.

Key words: computational linguistic, dependency parsing, natural language processing, text segmentation

АВТОМАТИЧЕСКАЯ ОБРАБОТКА ТЕКСТА: ДОСТУПНЫЙ ВЕБ-СЕРВИС ДЛЯ МОРФОЛОГИЧЕСКОЙ И СИНТАКСИЧЕСКОЙ АННОТАЦИИ РУССКИХ ТЕКСТОВ

Дроганова К. А. (kira.droganova@gmail.com),
Медянкин Н. С. (nikita.medyankin@gmail.com)

Школа лингвистики факультета филологии НИУ ВШЭ, Москва, Россия

Ключевые слова: компьютерная лингвистика, парсинг зависимостей, автоматическая обработка текста, сегментация текста

1. Introduction

The previous experiments on building automatic NLP pipeline for Russian have been conducted by Serge Sharoff (Sharoff, Nivre [8]). TreeTagger (Schmid, 1994 [7]) was used for morphological annotation. Lemmas were produced with CST-lemmatizer and a list of lemmatization rules. Syntactic parsing was conducted with Maltparser (Nivre et al, 2007 [4]) trained on Syntagrus [9]. The full pipeline was implemented and is available on the Internet. However, it is presented as a set of separate scripts, with quite a few specific actions and technical knowledge being required to use it, especially on Windows.

Inspired by Sharoff's experiments, we made it our goal to develop an easy-to-use web application built upon the pipeline utilizing the same idea but with different choices of tools involved. The main differences in pipeline are as follows.

First, segmentation is provided by the python3 module developed specifically for this task.

Second, tokenization, lemmatization, and morphological tagging are provided by Mystem (Segalovich, 2003 [6]) with additional corrections. Morphological information obtained from Mystem is disambiguated using TreeTagger with parameter model trained on disambiguated part of Russian National Corpus [2].

Third, a different parsing model for MaltParser is used.

Figure 1 provides the general scheme of the pipeline. The details are given in the sections below.

The paper is structured as follows: each of Sections 2, 3, and 4 provide detailed information regarding pipeline stages, which are text segmentation, morphology, and syntax respectively. Results regarding quality of each stage are provided in their respective sections. Section 5 is dedicated to the web interface. In Section 6, we conclude the article with future plans and provide some ideas on how to enhance the quality of the application in question.

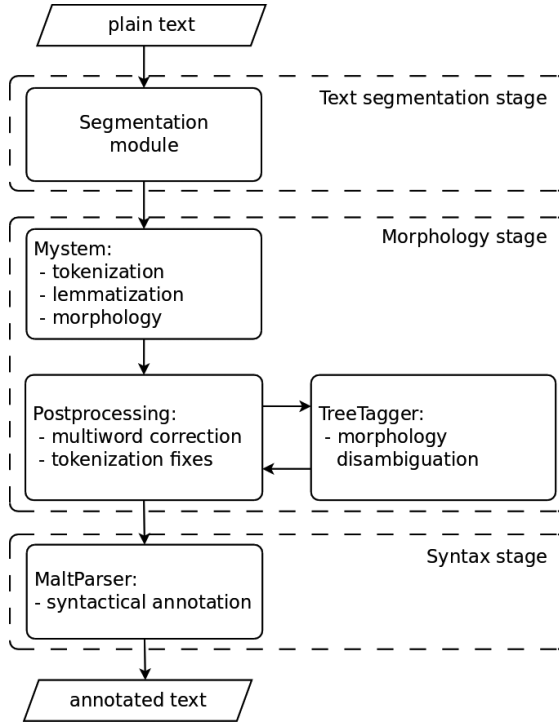

Fig. 1. NLP pipeline

2. Text Segmentation Stage

At the first stage, text segmentation module receives plain text as an input and determines ends of sentences using a rule-based approach. We consider correct segmentation an important requirement for quality syntax annotation, therefore an extensive effort has been put into development of our own segmentation module.

2.1. Rules

Rules for determining ends of sentences are applied at raw text level and are based on sequences of letters and terminal signs. The examples of core rules are listed below.

- Any number of dots, question marks, and exclamation marks in any combination followed by a capital letter is treated as the end of sentence, unless overridden by a specific rule.
- Semicolon is always considered the end of sentence.
- Colon is considered the end of sentence if followed by dash.
- The end of the line is always considered the end of sentence.

- The following combinations of letters and punctuation are never considered the end of sentence and override the rules stated above:
  - Abbreviation patterns:
    [.,]—The dot and comma sequence;
    [т.е]—The sequence of any lower-case/upper-case letters and the dot placed in between. For instance, e-mail and links are in agreement with this rule: example@.gmail.com, www.example.com;
    [т. е] —The sequence of any letter, and the dot placed after it, and the whitespace placed after it, and any lower-case letter;
    [П. И. Чайковский], [Чайковский П. И.]—A dot preceded by a single upper-case letter is never considered the end of sentence.
  - Quoted speech and explanation patterns:
    These patterns involve quotations and parentheses. Some examples are listed below:

(1) *«Прекрати!» — воскликнул Геннадий. / "Stop it!"—Gennady exclaimed.*

(2) *У них было пять двигателей: три бензиновых и два дизельных. / They had five engines: three of them were gasoline, the other two were diesel.*

2.2. Quality

The accuracy of text segmentation has been measured manually on a sample of 1,000 sentences of different genres and is 99.5%.

Testing has revealed a number of patterns that may cause wrong segmentation. Typically, these are addresses or amounts of money, which usually include sequences of numbers and abbreviations with dots. Consider an example:

(3) *25 руб. 33 коп.*

It is a number followed by a currency abbreviation ending with a dot, repeated two times. The sentence containing this sequence will be incorrectly split after *руб.*

Other common mismatches are caused by emoticon and emoji patterns. Segmentation module does not have specific rules for texts with erroneous punctuation and capitalization patterns such as blog posts because it was not intended as a tool for parsing specific genres.

2.3. Technical Details

The module is written in python3, rule patterns are determined using perl-style regular expressions. The input of segmentation step is plain text in utf-8 encoding, the output is plain text with special tag inserted at the end of each sentence.

3. Morphology Stage

At the second stage, tokenization, lemmatization, and morphological annotation are conducted. As was mentioned before, we use Mystem with some additional processing. The main arguments in favour of using Mystem are as follows.

First, Mystem was specifically designed for Russian, is based on extended Zaliznyak dictionary, and can also predict lemmas and grammatical information for unknown words with decent quality.

Second, Russian National Corpus [5], which is the major publicly available corpus resource for Russian, is morphologically annotated with Mystem. Therefore, one can use our web-application to produce morphologically and syntactically annotated corpus, which would extend RNC with new sentences.

3.1. Postprocessing

The major drawback of Mystem is that it is only equipped with lexical disambiguation feature, but not with morphological one, e.g., if a noun has homonymous Accusative and Nominative cases (which is typical for Russian inanimate nouns), both variants of annotation are provided by Mystem with no internal means to choose the correct one.

To address this problem, we use TreeTagger with parameter file trained on disambiguated part of RNC [2] to choose morphological annotation from those provided by Mystem.

This fix provides roughly 5% increase in morphological annotation accuracy, as opposed to just using the first morphological annotation available from Mystem.

We have also added postprocessing feature that is aimed to fix multiword expressions, e.g., *какой бы то ни было*. Naturally, Mystem divides those into separate tokens, which are then lemmatized and annotated separately. To resolve this issue, we had extracted a list of frequent multiword tokens from Syntagrus and created a dictionary. During postprocessing, the tokens are stacked up and given morphological annotation according to this dictionary.

A number of minor fixes mainly concerning correcting the tokenization of punctuation marks is also applied during postprocessing.

3.2. Quality

Quality has been measured on combined test set made from the development test set and the final test set. The same datasets were used for measuring Syntax quality. Section 4.4 provides detailed information regarding test sets. The resulting test set contains 34,668 sentences or 120,703 words.

20 most frequent mismatches are listed in Table 1. The first column shows the fraction of the tokens with the specified error in regard to all tokens in the test set, the second—the fraction of the tokens with the specified error in regard to all incorrectly annotated tokens. As can be clearly seen, the most common mismatches can be divided into four distinctive groups:

1. Wrong case for nouns and adjectives.
2. Brevis adjective annotated as adverb, e.g., *нужно, должно, известно, трудно, необходимо.*
3. Adverb mixed up with conjunction and vice versa with the major cases being *однако, как, когда, пока.*
4. Particle annotated as conjunction or adverb. The worst offender of the former is *и* with *то* as the distant second, and the latter are mostly presented by *уже, еще, почти* and *также.*

Accuracy of morphological annotation for full tag set measured in a strict sense (i.e., one missing or misplaced tag for a token is a miss, full match tag-by-tag is a hit) is 85.5%. Precision and recall of morphological annotation for full tag set measured in classical sense are 92.4% and 91.6% respectively.

3.3. Technical Details

The wrapper for Mystem and postprocessing module are written in python3. The version of Mystem used is 3.0 binary for Linux. The input of this step is plain text with the ends of sentences marked up, the output is conll file with empty positions for syntactic relations.

Table 1. Top 20 common mismatches in morphological annotation

total %	error %	annotated as	correct annotation
0.53%	3.67%	CONJ	PART
0.47%	3.23%	ADV	PART
0.31%	2.11%	ADV	A brev sg n
0.30%	2.08%	S sg m nom inan	S sg m acc inan
0.24%	1.62%	S sg m acc inan	S sg m nom inan
0.23%	1.56%	A pl nom plen	A pl acc inan plen
0.22%	1.49%	S sg n nom inan	S sg n acc inan
0.20%	1.35%	A sg m nom plen	A sg m acc inan plen
0.18%	1.23%	CONJ	ADV
0.14%	0.98%	S sg f loc inan	S sg f dat inan
0.13%	0.91%	A sg m acc plen	A sg m nom plen
0.13%	0.87%	A pl acc plen	A pl nom plen
0.12%	0.80%	S pl m nom inan	S pl m acc inan
0.11%	0.77%	ADV	CONJ
0.10%	0.72%	A sg n nom plen	A sg n acc plen
0.10%	0.72%	S sg n acc inan	S sg n nom inan
0.10%	0.71%	S sg m gen anim	S sg m acc anim
0.10%	0.70%	S sg f gen inan	S pl f nom inan
0.09%	0.65%	S pl f nom inan	S pl f acc inan
0.09%	0.64%	S sg f gen inan	S sg f loc inan

4. Syntax Stage

The third stage involves annotating syntactic layer. Syntactic annotation is provided by MaltParser working in parse mode.

4.1. Model

The parsing model has been trained on Syntagrus. SynTagRus was split into three parts: the training set (80%), the development test set (10%) and the final test set (10%). The original SynTagRus format (Iomdin et al. [10]) was converted into conll-file [11] using a conversion scheme.

It should be mentioned that the quality of the model utilized in current pipeline is 83.7% by LAS and 89.6% by UAS. Quality parameters have been measured by MaltEval [3].

4.2. Common Mismatches

20 most frequent syntax relation tag mismatches for cases when the head is annotated correctly are listed in Table 2. As with the morphology, the first column shows the fraction of the tokens with the specified error in regard to all tokens in the test set, the second—the fraction of the tokens with the specified error in regard to all tokens with correct head and incorrect syntax relation tag. As can be seen, the mismatches encountered are those that are quite usual for MaltParser models trained on Syntagrus.

Additionally, we have measured the impact of morphological annotation quality on syntactic annotation quality. Experiments have been conducted on 130 manually-annotated sentences. First, we obtained the syntactic annotation using manually-annotated morphological layer. Second, we obtained both annotations using the pipeline. Then we calculated the difference in syntactic annotation quality, which turned out to be 3.5% in favor of manually annotated morphology. It can be clearly seen that morphological annotation quality has quite an impact on syntactic annotation quality.

4.3. Technical Details

The input of the syntactic step is conll file with empty positions for syntactic relations, the output is conll file with both morphological and syntactic annotation.

5. Web Application

A deliberately simplistic web interface has been implemented on top of the pipeline, which allows the user to upload the text, wait for the pipeline to annotate it, and then download the results.

The web application is available for testing and unconditional use at http://web-corpora.net/wsgi3/ru-syntax/

Offline version is supplied as a python3 library with command line interface. The source code can be obtained from github at https://github.com/tiefling-cat/ru-syntax.

Table 2. Top 20 common mismatches in syntactic annotation

total %	error %	annotated as	correct annotation
0.47%	6.24%	1-компл	2-компл
0.47%	6.21%	1-компл	предик
0.44%	5.75%	квазиагент	1-компл
0.41%	5.42%	предик	1-компл
0.35%	4.69%	обст	1-компл
0.31%	4.11%	обст	2-компл
0.29%	3.88%	1-компл	обст
0.26%	3.44%	1-компл	квазиагент
0.25%	3.26%	обст	огранич
0.21%	2.77%	2-компл	1-компл
0.20%	2.60%	квазиагент	атриб
0.16%	2.07%	1-компл	атриб
0.15%	2.02%	атриб	1-компл
0.14%	1.91%	2-компл	обст
0.14%	1.82%	опред	квазиагент
0.10%	1.30%	опред	количест
0.09%	1.24%	опред	вспом
0.09%	1.14%	обст	присвяз
0.08%	1.06%	обст	3-компл
0.08%	1.03%	1-компл	3-компл

6. Conclusions

In this paper we have presented NLP web application for Russian texts that does not require any specific technical knowledge or software to use. This way linguists conducting fundamental research on some collection of raw text would be able to concentrate on the research itself, not on looking for the tools to annotate the corpus and desperately trying to get them to work. Due to its usage of the same morphological tagset as Russian National Corpus, one can possibly use our application to obtain a morphologically and syntactically pre-annotated corpus of Russian texts compatible with RNC.

In our future work we are planning to concentrate on the segmentation quality. It should be tested more closely on a larger amount of testing data. Since our application should be able to successfully process texts of any origin, emoticon and emoji processing rules should be added due to their frequency on the Internet. As future work, we also consider experiments on improving morphological annotation quality, due to both its importance on itself and its significant impact on the aggregated result.

It should also be noted that the authors of this work are unaware of any other tool offering NLP pipeline for Russian going from plain text to syntactic annotation working out of the box and at the same time being free to use. The only one that might be comparable is the pipeline put together by Sharoff himself in 2011, but it works out of the box only for Linux-based systems, and we could find no reported results regarding its overall accuracy.

Acknowledgements

The authors are grateful for computational capabilities provided by Andrey Kutuzov and mail.ru group and appreciate support and feedbacks from Olga Lyashevskaya from School of Linguistics, Faculty of Humanities, Higher School of Economics, Moscow.

Reference

1. *Droganova K.* (2015), Building a Dependency Parsing Model for Russian with MaltParser and MyStem Tagset, Proceedings of the AINL-ISMW FRUCT, Saint-Petersburg, Russia, 9–14 November 2015, ITMO University, FRUCT Oy, Finland.
2. *Fenogenova A., Kayutenko D., Dereza O.* Mystem+, available at: http://web-corpora.net/wsgi/mystemplus.wsgi/mystemplus/
3. *Nilsson J.* (2014), User Guide for MaltEval 1.0 (beta), available at: http://www.maltparser.org/malteval.html
4. *Nivre J., Hall J., Nilsson J., Chanev A., Eryigit G., Kübler S., Marinov S., Marsi E.* (2007), MaltParser: A language independent system for data-driven dependency parsing, Natural Language Engineering, Vol. 13, pp. 95–135.
5. *Russian National Corpus:* http://www.ruscorpora.ru/en/index.html
6. *Segalovich I.* (2003), A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine, MLMTA-2003, available at: https://tech.yandex.ru/mystem/
7. *Schmid H.* (1994), Probabilistic Part-of-speech Tagging Using Decision Trees, International Conference on New Methods in Language Processing
8. *Sharoff S., Nivre J.* (2011), The proper place of men and machines in language technology. Processing russian without any linguistic knowledge, Computational Linguistics and Intelligent Technologies: Proceedings of the International Conference Workshop "Dialogue 2011" [Komp'yuternaya Lingvistika i Intellektual'nye Tekhnologii: Trudy Mezhdunarodnoy Konferentsii "Dialog 2011"], Bekasovo, Vol. 10 (17), pp. 657–670.
9. *Syntagrus Instruction,* available at: http://www.ruscorpora.ru/instruction-syntax.html
10. *Iomdin, Leonid, Petrochenkov, Vyacheslav, Sizov, Viktor, Tsinman Leonid* (2012) ETAP parser: state of the art. In Computational linguistics and intellectual technologies. Proceedings of International Workshop Dialogue 2012. Vol. 11 (18), 2012. Moscow: RGGU, pp. 830–848.
11. *Depparse Wiki:* http://depparse.uvt.nl/DataFormat.html