

Dependency Treebank for Russian: Concept, Tools, Types of Information

Igor BOGUSLAVSKY, Svetlana GRIGORIEVA,
Nikolai GRIGORIEV, Leonid KREIDLIN, Nadezhda FRID

Laboratory for Computational Linguistics
Institute for Information Transmission Problems

Russian Academy of Sciences

Bolshoi Karetnyi per. 19, 101447 Moscow – RUSSIA

{bogus, sveta, grig, lenya, nadya}@iitp.ru

Abstract

The paper describes a tagging scheme designed for the Russian Treebank, and presents tools used for corpus creation.

1. Introductory Remarks

The present paper describes a project aimed at developing the first annotated corpus of Russian texts. Large text corpora have been used in the computational linguistics community long enough: at present, over 20 large corpora for the main European languages are available, the largest of them containing hundreds of millions of words (Language Resources (1997); Marcus, Santorini, and Marcinkiewicz (1993); Kurohashi, Nagao (1998)). So far, however, no annotated corpora for Russian have been developed. To the best of our knowledge, the present project is the first attempt to fill the gap.

Different tasks require different annotation levels that entail different amount of additional information about text structure. The corpus that is being created in the framework of the present project consists of several subcorpora that differ by the level of annotation. The following three levels are envisaged:

- *lemmatized texts*: for every word, its normal form (*lemma*) and part of speech are indicated;
- *morphologically tagged texts*: for every word, a full set of morphological attributes is specified along with the lemma and the part of speech;
- *syntactically tagged texts*: apart from the full morphological markup at the word level, every sentence has a syntax structure.

We annotate Russian texts with *dependency structures* – a formalism that is more suitable for Slavonic languages with their relatively free word order. The structure not only contains information on which words of the sentence are syntactically linked, but also relegates each link to one of the several dozen syntactic types (at present, we use 78 syntactic relations). This formalism ensures a more complete and informative representation than any other syntactically annotated corpus. This is a major innovation, since the majority of syntactically annotated corpora, both those already available and under construction, represent the syntactic structure by means of constituents.

The closest analogue to our work is the Czech annotated corpus collected at Charles University in Prague – see Hajicova, Panevova, Sgall (1998). In this corpus, the syntactic data are also expressed in a dependency formalism, although the set of syntactic functional relations is much smaller as it only has 23 relations

In what follows, we describe the types of texts used to create the corpus (Section 2), markup format (Section 3), annotation tools and procedures (Section 4), and types of linguistic data included in the markup (Section 5).

2. Source text selection

The well-known Uppsala University Corpus of contemporary Russian prose, totalling ca. 1,000,000 words, has been chosen as the primary source for our work. The Uppsala Corpus is well balanced between fiction and journalistic genre, with a smaller percentage of scientific and popular science texts. The Corpus includes samples of contemporary Russian prose, as well as excerpts from newspapers and magazines of recent decades, and gives a representative coverage of

written Russian in modern use. Conversational examples are scarce and appear as dialogues inside fiction texts.

3. Markup format

The design principles were formulated as follows:

- “layered” markup – several annotation levels coexist and can be extracted or processed independently;
- incrementality – it should be easy to add higher annotation levels;
- convenient parsing of the annotated text by means of standard software packages.

The most natural solution to meet this criteria is an XML-based markup language. We have tried to make our format compatible with TEI (Text Encoding for Interchange, see TEI Guidelines (1994)), introducing new elements or attributes only in situations where TEI markup does not provide adequate means to describe the text structure in the dependency grammar framework.

Listed below are types of information about text structure that must be encoded in the markup, and relative tags/attributes used to bear them.

a) Splitting of text into sentences. A special container element **<S>** (available in TEI) is used to delimit sentence boundaries. The element may have an (optional) **ID** attribute that supplies a unique identifier for the sentence within the text; this identifier may be used to store information about extra-sentential relations in the text. It may also have a **COMMENT** attribute, used by linguists to store observations about particular syntactic phenomena encountered in the sentence;

b) Splitting of sentences into lexical items (words). The words are delimited by a container element **<W>**. Like sentences, words may have a unique **ID** attribute that is used to reference the word within the sentence;

c) Ascribing morphological features to words. Morphological information is ascribed to the word by means of two attributes born by the **<W>** tag:

LEMMA – a normalized word form;

FEAT – morphological features.

d) Storing information about the syntax structure. To annotate the information about syntactic

dependencies, we use two other attributes in the **<W>** element:

DOM – the **ID** of the master word;

LINK – syntactic function label.

There are also special provisions in the formalism to store auxiliary information, e.g. multiple morphological analyses and syntax trees. They are expected to disappear from the final version of the corpus.

4. Annotation tools and procedures

The procedure of corpus data acquisition is semi-automatic. An initial version of markup is generated by a computer using a general purpose morphological analyzer and syntax parser engine; after that, the results of the automatic processing are submitted to human post-editing. The analysis engine (morphology and parsing) is based upon the ETAP-3 machine translation engine – see Apresjan et al. (1992, 1993).

To support the creation of annotated data, a set of tools was designed and implemented. All tools are Win32 applications written in C++. The tools available are:

- a program for sentence boundaries markup, called **Chopper**;
- a post-editor for building, editing and managing syntactically annotated texts – **Structure Editor** (or **StrEd**).

The amount of manual work required to build annotations depends on the complexity of the input data. **StrEd** offers different options for building structures. Most sentences can be reliably processed without any human intervention; in this case, a linguist should look through the processing result and confirm it. If the structure contains errors, the linguist can edit it using a user-friendly graphical interface (see screenshots below). If the errors are too many or no structure could be produced, the linguist may use a special *split-and-run mode*. This mode includes manual pre-chunking of the input phrase into pieces with a more transparent structure and applying the analyzer/parser to every chunk. Then the linguist must manually link the subtrees produced for every chunk into a single structure.

If the linguist has encountered a very peculiar syntactic construction so that he/she is uncertain

about the correct structure, he/she may mark as “doubtful” the whole sentence or single words whose functions are not completely clear. The information will be stored in the markup, and **StrEd** will visualize the respective sentence as one in need for further editing.

Fig.1 presents the main dialog window for editing sentence properties. An operator can edit the markup directly, or edit single properties using a graphical interface. The source text under analysis is written in an edit window in the top: *Xotja pis'mo ne bylo podpisano, ja mgnovenno dogadalsja, kto ego napisal* [Although the letter was not signed, I instantly guessed who had written it]. The information about single words is written into a list: e.g. the first word *xotja* [although] has an identifier **ID="1"**; the lemmatized form is *XOTJA*; its feature list consists of a single feature – a part-of-speech characteristic (it is a conjunction); the word depends on a word with **ID="8"** by an adverbial

relation (link type is "**adverb**"). By double-clicking an item in the word list or pressing the button, a linguist can invoke dialog windows for editing properties of single words. However, the most convenient way of editing the structure consists in invoking a **Tree Editor** window, shown in Fig. 2 with the same sentence as in the previous picture.

The Tree Editor interface is simple and natural. Words of the source sentence are written on the left, their lemmas are put into gray rectangles, and their morphological features are written on the right. The syntactic relations are shown as arrows directed from the master to the slave; the link types are indicated in rounded rectangles on the arcs. All text fields except for the source sentence are editable in-place. Moreover, one can drag the rounded rectangles: dropping it on a word means that this word is declared a new master for the word from which the rectangle was dragged. A single right-button click on the lemma rectangle

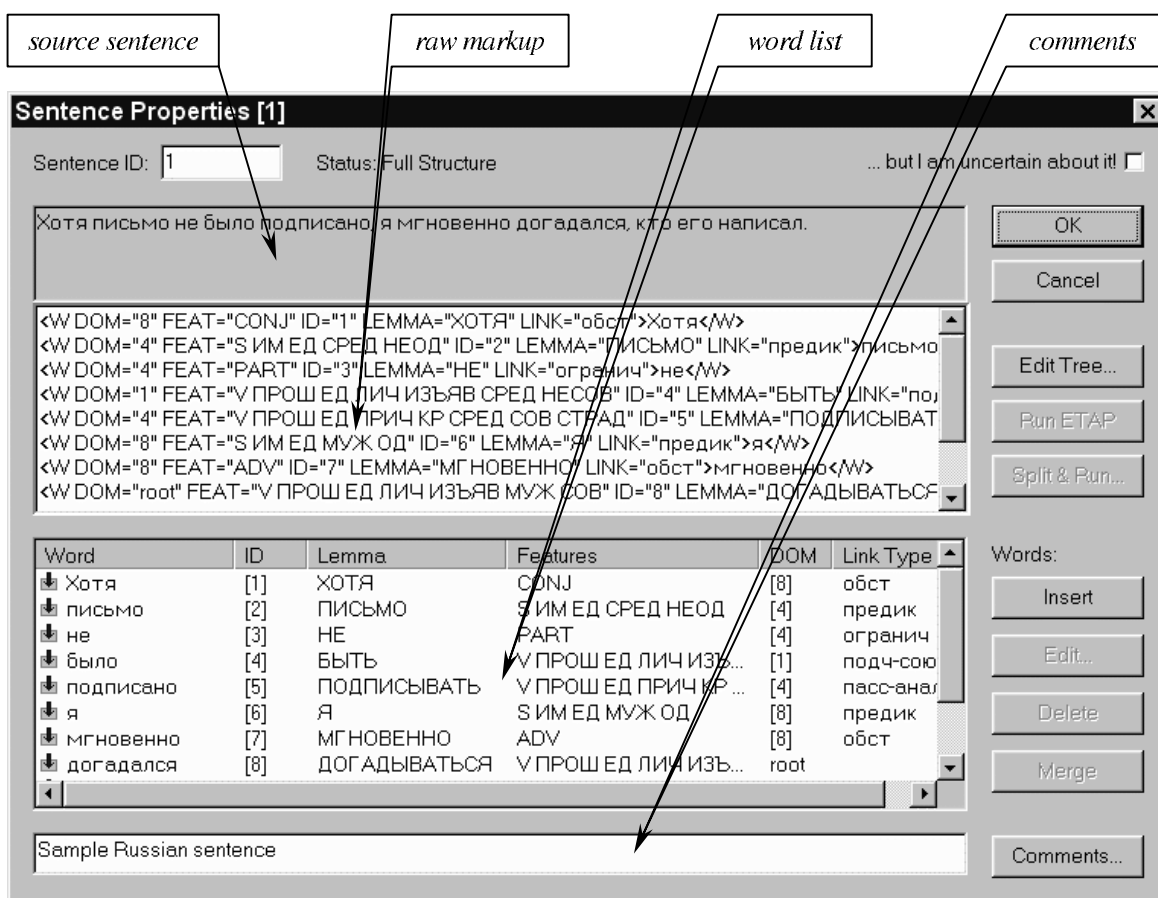


Figure 1. Sentence Properties dialog in StrEd.



Figure 2. Tree Editor dialog in StrEd.

brings out the word properties dialog. All colors, sizes and fonts are customizable.

5. Types of linguistic information by level

Morphology information

The morphological analyzer ascribes features to every word. The feature set for Russian includes:

part of speech, animateness, gender, number, case, degree of comparison, short form (of adjectives and participles), representation (of verbs), aspect, tense, person, voice.

Syntax information

As we have already mentioned, the result of the parsing is a tree composed of links. Links are binary and oriented; they link single words rather than syntactic groups. For every syntactic group, one word (*head*) is chosen to represent it as a slave in larger syntactic units; all other members of the group become slaves of the head.

In a typical case, the number of nodes in the syntactic tree corresponds to the number of word tokens. However, several exceptional situations occur in which the number of nodes may be less or even greater than the number of word tokens. The latter case is especially interesting. We postulate such a description in the following cases:

- Copulative sentences in the present tense where the auxiliary verb can be omitted. This is treated as a special “zero-form” of the copula, e.g. *On – uchitel’* [*He is a teacher*, lit.

He – teacher]. The copula should be introduced in the syntactic representation.

- Elliptical constructs (omitted members of contrasted coordinative expressions), like in *Ja kupil rubashku, a on galstuk* [*I bought a shirt, and he bought a necktie*, lit. *I bought a shirt, and he a necktie*].

The latter type of sentences should be discussed in more detail. Elliptical constructions are known to be one of the toughest problems in the formalization of natural language syntax. In our corpus, we decided to reconstruct the omitted elements in the syntactic trees, marking them with a special “phantom” feature. In the above example, a phantom node is inserted into the sentence between the words *on* ‘he’ and *galstuk* ‘necktie’. This new node will have a lemma *POKUPAT’* [*BUY*] and will bear exactly the same morphological features as the wordform *kupil* [*bought*] physically present in the sentence, plus a special “phantom” marker. In certain cases, the feature set for the phantom may differ from that of the prototype, e.g. in a slightly modified phrase *Ja kupil rubashku, a ona galstuk* [*I bought a shirt, and she (bought) a necktie*] the phantom node will have the feminine gender, as required by the agreement with the subject of the second clause. Most real-life elliptical constructs can be represented in this way.

The inventory of syntactic relationship types generated by the ETAP-3 system is vast enough: at present, we count 78 different syntactic function types. All relationships are divided into 6

major groups: **actant**, **attributive**, **quantitative**, **adverbial**, **coordinative**, **auxiliary**.

For readers' convenience, we will give equivalent English examples:

Actant relationships link the predicate word to its arguments. Some examples ([X] – master, [Y] – slave):

predicative – *Pete* [Y] *reads* [X];
completive (1, 2, 3) – *translate* [X]
the book [Y, 1-compl]
from [Y1, 2-compl] *English*
into [Y2, 3-compl] *Russian*

Attributive relationships often link a noun to a modifier expressed by an adjective, another noun, a participle clause, etc:

relative – *The house* [X] *we live*[Y] *in*.

Quantitative relationships link a noun to a word with quantity semantics, or two such words one to another:

quantitative – *five* [Y] *pages* [X];
auxiliary-quantitative – *thirty* [Y] *five* [X];

Adverbial relationships link the predicate word to various adverbial modifiers:

adverbial – *come* [X] *in the evening* [Y];
parenthetic – *In my opinion* [Y], *that's* [X] *right*.

Coordinative relationships serve for clauses coordinated by conjunctions:

coordinative – *buy apples* [X] *and pears*[Y] ;
coordinative-conjunctive – *buy apples*
and [X] *pears* [Y].

Auxiliary relationships typically link two elements that form a single syntactic unit:

analytical – *will* [X] *buy* [Y];

The list of syntactic relations is not closed. The process of data acquisition brings up a variety of rare syntactic constructions, hardly covered by traditional grammars. In some cases, this has led to the introduction of new syntactic link types in order to reflect the semantic relation between single words and make the syntactic structure unambiguous.

Conclusion

Corpus creation is not yet completed: at present, the full syntactic markup has been generated for

4,000 sentences (55,000 words), which constitutes 30% of the total amount planned. Our approach permits to include all information expressed by morphological and syntactic means in contemporary Russian. We expect that the new corpus will stimulate a broad range of further investigations, both theoretical and applied.

We plan to make the corpus available via ELRA framework after completion. Samples of tagged text, documentation and structure editing tools will be available for download from our site: <http://proling.iitp.ru/Corpus/preview.zip>.

Acknowledgements

This work is supported by Russian Foundation of Fundamental Research, grant No. 98–0790072.

References

- Apresjan Ju.D., Boguslavskij I.M., Iomdin L.L., Lazurskij A.V., Sannikov V.Z. and Tsinman L.L. (1992). *The linguistics of a Machine Translation System*. *Meta*, 37 (1), pp. 97–112.
- Apresjan Ju.D., Boguslavskij I.M., Iomdin L.L., Lazurskij A.V., Sannikov V.Z. and Tsinman L.L. (1993). *Système de traduction automatique ETAP*. In: *La Traductive*. P.Bouillon and A.Clas (eds). Les Presses de l'Université de Montréal, Montréal.
- Hajicova E., Panevova J., Sgall P. (1998). *Language Resources Need Annotations To Make Them Really Reusable: The Prague Dependency Treebank*. In: *Proceedings of the First International Conference on Language Resources & Evaluation*, pp. 713–718.
- Kurohashi S., Nagao M. (1998). *Building a Japanese Parsed Corpus while Improving the Parsing System*. In: *Proceedings of the First International Conference on Language Resources & Evaluation*, pp. 719–724
- Language Resources (1997). In: *Survey of the State of the Art in Human Language Technology*. Eds. G. B. Varile, A. Zampolli, *Linguistica Computazionale*, vol. XII–XIII, pp. 381–408.
- Marcus M. P., Santorini B., and Marcinkiewicz M.-A. (1993). *Building a large Annotated Corpus of English: The Penn Treebank*. *Computational Linguistics*, Vol. 19, No. 2.
- TEI Guidelines (1994). *TEI Guidelines for Electronic Text Encoding and Interchange (P3)*. URL: <http://etext.lib.virginia.edu/TEI.html>