Development of a Dependency Treebank for Russian and its Possible Applications in NLP

Igor BOGUSLAVSKY, Ivan CHARDIN, Svetlana GRIGORIEVA, Nikolai GRIGORIEV, Leonid IOMDIN, Leonid KREIDLIN, Nadezhda FRID

Laboratory for Computational Linguistics
Institute for Information Transmission Problems
Russian Academy of Sciences
Bolshoj Karetnyj per. 19, 101447, Moscow – RUSSIA
{bogus, ic, sveta, grig, iomdin, 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 for quite a long time now; 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)). For Russian, annotated corpora had been nonexistent until 2000 when the first part of the corpus reported here was compiled (Boguslavsky et al., 2000). Since then, Russian corpus linguistics has been evolving rapidly, and several groups of researchers announced their intent to create corpora. Of these, the CLD-MSU Corpus project looks particularly promising. It aims at building a morphologically tagged corpus, and an upgrade to syntactic annotation is envisaged in the future even though it is not pursued at the present stage (Sichinava, 2001).

Different tasks require different annotation levels that entail different amount of information about text structure. The corpus that is being created in the framework of the project under discussion consists of several subcorpora that differ in 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, along with the lemma and the part of speech, a full set of inflectional morphological attributes is specified;
- *syntactically tagged texts:* apart from the full morphological markup at the word level, every sentence is assigned a syntactic structure.

We annotate Russian texts with *dependency structures* – a formalism which we consider more suitable for Slavonic languages with their relatively free word order than constituent structures. The structure not only contains information as to 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 is an important feature, 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 Prague Dependency Treebank (PDT) – an annotated corpus of Czech collected at Charles University in Prague (see Hajicova, Panevova, Sgall, 1998). In this corpus, the syntactic data are also expressed in the dependency formalism, although the inventory of syntactic functional relations is much smaller than ours as it only has 23 relations. Our corpus therefore gives a more fine-grained representation of syntactic phenomena. On the other hand, Czech researchers made an extremely interesting attempt to incorporate into their annotation information on discourse structure (topic-focus opposition) (Bemova et al., 1999).

Besides PDT, several other corpus-related projects use some kind of dependency structures; worth noticing are NEGRA for German (Brants et al, 1999) and Alpino Dependency Treebank for Dutch (Van der Beek et al., 2001).

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 has been chosen as the primary source for the first part of our corpus, which has already been completed. This part contains about 10,000 sentences. 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 the last few decades of the 20th century, and gives a representative coverage of written Russian in modern use. Conversational examples are scarce and appear as dialogues inside fiction texts.

The second part of the corpus consists of several hundred short texts published in 2001-2002 on various Internet news portals. The bulk of the texts come from the following newswires: www.yandex.ru, www.rbc.ru, www.polit.ru, www.lenta.ru, www.strana.ru, www.news.ru. Each text is a small story (up to 30 sentences) about a single event. Their themes include political, financial, cultural, and sports news, both domestic and international; and a certain amount of texts deal with hi-tech achievements. We have done our best to make source text selection a representative sample of Internet news delivery in Russian.

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 these 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 the respective tags/attributes used to carry this information.

3.1. Splitting 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 notes and observations on about particular syntactic phenomena encountered in the sentence;

3.2. Splitting 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 refer to the word within the sentence;

3.3. Assigning Morphological Features to Words. Morphological information is ascribed to the word by means of two attributes attached to the **<W>** tag: **LEMMA** – normalized word form and **FEAT** – list of morphological features.

3.4. Storing Information about Syntactic Structure. To annotate the information about syntactic dependencies, we use two other attributes attached to the **<W>** element: **DOM** – the **ID** of the word on which **W** depends and **LINK** – syntactic function label.

The formalism has special provisions to store auxiliary information, e.g. multiple morphological analyses and syntactic trees. They will not appear in 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 as discussed in Apresjan et al. (1992, 1993).

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

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

The amount of manual labor 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 which case, a linguist should only look through the result of the processing and endorse 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 resort to a special **split-and-run mode.** This mode involves manual pre-chunking of the input sentence into such pieces that have a more transparent structure and applying the analyzer/parser to every chunk in turn. Then the linguist must manually integrate the subtrees produced for every chunk into a single tree structure.

If the linguist has come across an especially difficult syntactic construction so that he/she is uncertain about what the adequate structure is, he/she may mark as "doubtful" the whole sentence or else 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 needing further editing.

Fig. 1 presents the main dialog window for editing sentence properties. An operator can edit the markup directly in any text editor, or edit single properties using a graphical interface. The source text under analysis is written in the top line of the edit window: *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 particular 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 the 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 placed 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 pops up the word properties dialog. All colors, sizes and fonts are customizable.

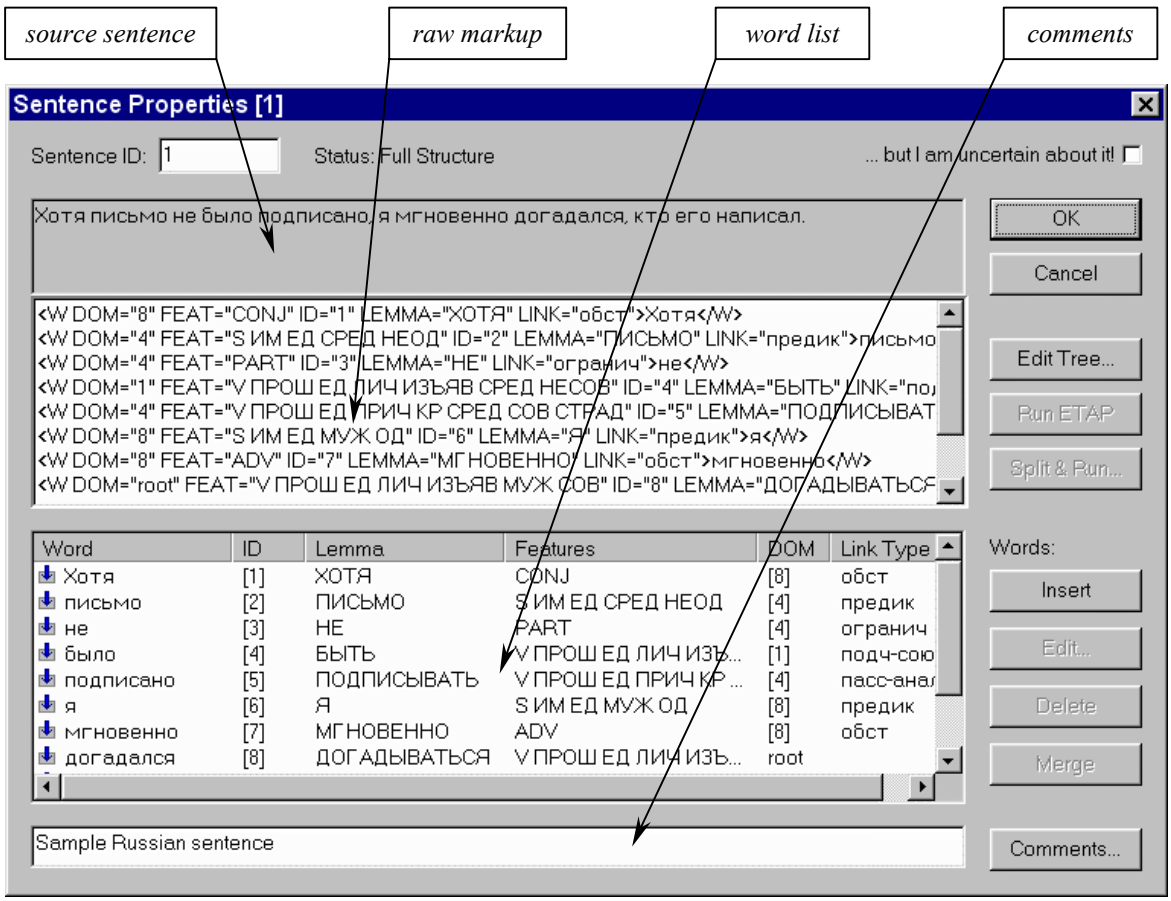

Figure 1. Sentence Properties dialog in **StrEd**.

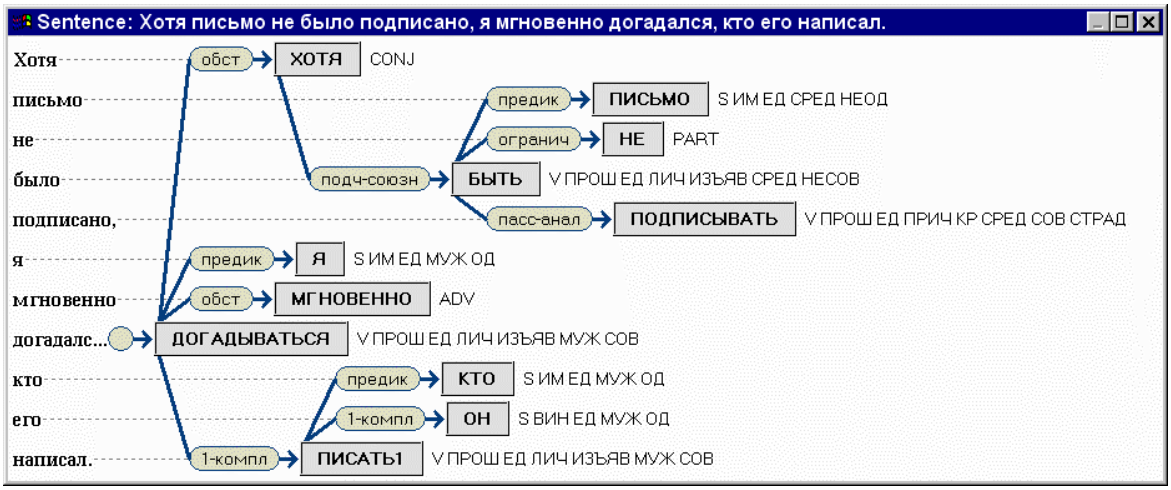

Figure 2. Tree Editor dialog in **StrEd**.

5. Types of Linguistic Information by Level

5.1. Morphological 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 (for adjectives and participles), representation (of verbs), aspect, tense, person, and voice.

5.1. Syntactic Information

As we have already mentioned , the result of the parsing is a tree composed of links as leaves and words as nodes. All the 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 either less or greater than the number of word tokens. The latter case is especially interesting. We postulate such a description in the following cases:

a) 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.

b) Elliptical constructions (e.g. 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 relations are divided into 6 major groups: **actant**, **attributive**, **quantitative**, **adverbial**, **coordinative**, **auxiliary**.

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

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

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

Attributive relations 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 relations link a noun to a quantifier or numeral, or two such words together:

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

Adverbial relations link the predicate word to various adverbial modifiers:

adverbial – *He came* [X] *every evening* [Y];
parenthetic – *In* [Y] *my opinion*, he *is* [X] *right*.

Coordinative relations serve phrases and clauses coordinated by conjunctions:

coordinative – *buy apples* [X_1] , pears [Y_1= X_2] *and* [Y_2] *apricots;*
coordinative-conjunctive – *buy apples and* [X] *pears* [Y].

Auxiliary relations typically link two elements that form a single syntactic unit (e.g. an analytical verb form):

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, if at all, covered by traditional grammars. In some cases, this has led to the introduction of new syntactic link types in order to reflect a partucular relation between single words and make the syntactic structure unambiguous.

6. Application of the Tagged Corpus in NLP

The first type of research application on which we have started to test the annotated corpus is resolution of syntactic ambiguity in the course of Russian parsing as part of ETAP-3 Russian-to-English machine translation. Within the parser, an additional filter has been created that assigns weights to all potential subtrees of the sentence processed that consist of two to four nodes (so-called N-grams of the 1st, 2nd and 3rd order) based on their relative occurrence in the dependency trees belonging to the Treebank. (For details, see Chardin 2001). Weights of the concurrent subtrees are compared with the existing priority values of individual syntactic links in the operational space of the parser and modify these values accordingly, which eventually helps create a more adequate tree structure. First results are promising; a detailed report on the ongoing experiments is being prepared. In fact, the results of such experiments are reusable in the creation of the Treebank itself, since new automatically derived parses that take into account the subtree weights are likely to show a better conformity with the previously produced corpus and will require less manual editing.

Conclusion

Corpus creation is not yet completed: at present, the full syntactic markup has been generated for 12,000 sentences (180,000 words), which constitutes 50% 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 research and development projects..

We plan to make the corpus publicly available after completion.

Acknowledgement

This work has been supported by the Russian Foundation of Fundamental Research with a grant No. 01-07-90405.

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*. // *La Traductique.* P.Bouillon and A.Clas (eds). Les Presses de l'Université de Montréal, Montréal.

Boguslavsky I.M., Grigorieva S.A., Grigoriev N.V, Kreidlin L.G, Frid N.E. (2000). *Dependency Treebank for Russian: Concepts, Tools, Types of Information.* // Proceedings of the 18th Conference on Computational Linguistics. Vol 2, 987-991, Saarbrücken.

Chardin I.S. (2001). *Using a tagged corpus to resolve syntactic ambiguity in the ETAP-3 Linguistic Processor.* // Proceedings of the 2nd All-Russian Conference "Theory and Practice of Speech Resources" (ARSO-2001). Moscow State University, Moscow. [in Russian]

Hajicova E., Panevova J., Sgall P. (1998). *Language Resources Need Annotations To Make Them Really Reusable: The* Prague *Dependency Treebank.* // 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.* // Proceedings of the First International Conference on Language Resources & Evaluation, pp. 719–724

Language Resources (1997). 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

Brants Th., Skut W., and Uszkoreit H. (1999). *Syntactic annotation of a German newspaper corpus*. // Proceedings of the ATALA Treebank Workshop, pp. 69-76, Paris, France.

Van der Beek L., Bouma G., Malouf R., van Noord G. (2001). The Alpino Dependency Treebank, In: Proceedings of LINC2001.

Sichinava, D.V. (2001). *On the problem of building Russian linguistic* corpora *for the Internet*. URL: www.mccme.ru/ling/mitrius/article.html [in Russian]

Bemova A., Hajic J., Hladka B., Panevova J. (1999). Morphological and Syntactic Tagging of the Prague Dependency Treebank. URL: http://citeseer.nj.nec.com/296119.html