СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР СИСТЕМЫ ЭТАП: СОВРЕМЕННОЕ СОСТОЯНИЕ.

Иомдин Л. Л. (iomdin@iitp.ru), **Петроченков В. В.** (petrochenkvov@iitp.ru), **Сизов В. Г.** (sizov@iitp.ru), **Цинман Л. Л.** (cinman@iitp.ru)

Институт проблем передачи информации РАН им A. A. Харкевича, Москва

Излагается современное состояние синтаксического анализатора ЭТАП-3, участвовавшего в соревновании русских парсеров. Приводятся сведения об основных лингвистических ресурсах, участвующих в работе анализатора, об алгоритме его работы, а также о приложениях, в которых используется этот анализатор, включая систему машинного перевода, систему создания синтаксически размеченного корпуса текстов и гибридную систему русского речевого синтеза. Особое внимание уделяется конкретным научным подходам и решениям, обусловливающим особенности функционирования анализатора, в частности, методам разрешения лексической и синтаксической неоднозначности.

Ключевые слова: синтаксический анализатор, комбинаторный словарь, синтагмы, размеченные корпуса текстов

ETAP PARSER: STATE OF THE ART

lomdin L. (iomdin@iitp.ru),
Petrochenkov V. (petrochenkvov@iitp.ru),
Sizov V. (sizov@iitp.ru),
Tsinman L. (cinman@iitp.ru)

Institute of Information Transmission Problems (Kharkevich Institute), Russian Academy of Sciences

The state of the art of the ETAP-3 syntactic parser, which took part in a recent competition of Russian parsers, is presented. The paper gives an outline of the main linguistic resources involved in the parser's operation, describes the main features and steps of the algorithm, and briefly discusses the applications in which the parser is used, including a machine translation system, a software environment for the creation of a syntactically tagged corpus of Russian, and a hybrid system of Russian speech synthesis. Special attention is given to concrete scientific approaches and solutions that determine the functioning of the parser, including methods of lexical and syntactic disambiguation.

Key words: parser, combinatorial dictionary, syntagm, tagged text corpora

1. General Information

The syntactic parser presented here is the central component of the ETAP-3 multipurpose linguistic processor, designed and developed at the Laboratory of computational linguistics of the Institute for Information Transmission Problems in Moscow.² It has two major options operating on very similar (although not identical) principles: the parser of Russian and the parser of English. In what follows, only the parser of Russian will be described.

The parser (to be henceforth called ETAP, for short) is rule-based, with some statistical components incorporated recently.

ETAP processes the text sentence by sentence and has several modes of operation:

• fully automatic mode, applied by default: in this case, only one syntactic structure is built for any sentence processed;

The author is grateful to the Russian Foundation of Basic Research, who supported the research upon which this paper is based with grants No.10-06-00478-a and 11-06-00405-a, and to the Presidium of the Russian Academy of Sciences, who supported this study with a Basic Studies Programme on Corpus Linguistics.

² Earlier versions of the parser, as well as individual aspects of its performance and maintenance. were described in detail in Apresjan et al 1989,1992, 2003, Boguslavsky et al 2008, 2011.

- multiple parsing mode, in which the user may instruct the system to build, for an ambiguous sentence, several syntactic structures or even all possible structures:
- interactive mode, in which ETAP stops at certain points of the algorithm if it encounters an ambiguous lexical unit or syntactic construction. In this case, the user is asked to prompt the system for a morphological, lexical, or syntactic interpretation of the ambiguous element of the sentence and in this way direct the algorithm to take some concrete path.

The ETAP parser is primarily aimed at processing texts of neutral genres (journalism, popular science texts, news messages and the like). It cannot be used to adequately handle colloquial speech, fiction, or poetry, as well as "dirty" texts full of tables, lists, or indexes, as well as texts that are essentially deviant from the Russian literally norm.

1.1. Major Linguistic Conventions

The linguistic formalism used in ETAP is dependency grammar, and the structures produced are dependency tree structures. To a large degree, it is based on the Meaning ⇔ Text linguistic theory by Igor Mel'čuk, particularly on its surface syntactic component (see e.g. Mel'čuk 1974/1999). ETAP operates with written text, constructing a dependency tree for each sentence of it in turn. The tree, in accordance with its definition, has a single head which dominates, directly or indirectly, all other nodes (=leaves) of the tree. As a rule, every node corresponds to one word of the sentence. Importantly, punctuation marks do not constitute any nodes (they are generally attached to the words preceding them). In certain cases a node can correspond to a string of words, which is treated as an indivisible word for linguistic and/or algorithm optimization reasons: these are mostly compound prepositions or adverbs that never, or almost never, appear as sequences of separate words, such as no κραŭμεŭ μερε 'at least', εο чπο δω πο ни стало 'in any case', несмотря на 'in spite of' etc. Such exceptions are few, and these sequences are introduced into the dictionary sparingly and on an individual basis.

The arcs of the tree are labeled with names of surface syntactic relations (SyntR). These names indicate the different types of syntactic links between the words. In the current version of the parser, about 70 SyntRs are used. To give a few basic examples,

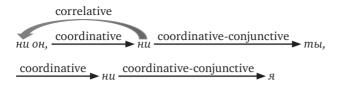
- the link between a predicate, expressed by a finite verb, which is the head, and its subject, which is the dependent, as in *omeų* ← *nonyчun* 'father received', is represented with **predicative SyntR**;
- the link going from a predicate word (verb, noun, adjective, or adverb) to the word instantiating its first complement, as in *получил* → *письмо* 'received a letter', *получение* → *письма* 'reception of a letter', *экивалентный* → *отказу* 'equivalent to a refusal', *вглубь* → *леса* 'deep into the wood' etc. is represented by the 1st completive SyntR;

- the link attaching the nominal part of the predicate to the copula verb, as in был → зол 'was angry' or будучи → учителем 'being a teacher' is represented by the copulative SyntR:
- the link connecting a noun and its adjectival modifier, as in заказное ← письмо 'registered letter', is represented by the **modificative** SyntR;
- the **adverbial SyntR** is used to represent modifiers of verbs expressed by adverbs or prepositional phrases, as in *неожиданно* ← *получил* 'unexpectedly received' or *получил* → в понедельник 'received on Monday';
- analytic forms of words (future tense or subjunctive mood of verbs and comparative degrees of adjectives and adverbs) are considered as syntactic constructions and represented with the help of the **analytic SyntR** (будет → читать 'will read', читал → бы 'would read', более ← интересный 'more interesting');
- the link between a noun and a numeral that refers to it is represented with **quantitative SyntR**, as in ∂ва ← стола 'two tables', пятерыми ← лингвистами 'by five linguists'. Importantly, the link always points to the numeral³;
- composite words like розовощёкий 'pink-cheeked' or столятидесятилетие 'one hundred and fiftieth anniversary', if they are not present in the dictionary, are segmented into parts linked consecutively from right to left with the help of composite SyntR: розово ← щекий, сто ← пятидесяти ← летний⁴;
- coordination is rendered on a par with subordination; coordination strings are
 presented in such a way that the first conjunct is the head on which the second
 conjunct depends and so on; a coordinating conjunction is subordinated by the
 conjunct preceding it. In most cases, two syntactic relations are used: the coordinative SyntR that links the neighboring conjuncts from left to right, and the
 coordinative-conjunctive SyntR that appends a conjunct to the left-adjacent
 conjunction, as in

'students and professors'. Special provision is made for correlative conjunctions like $\mu u \dots \mu u$ 'neither \dots nor', $\pi u \omega$ 'neither \dots nor', $\pi u \omega$ 'both \dots and' etc.: all members of the series of conjunctions except the leftmost one are linked from left to right, while the leftmost conjunction is dominated by the second one by **correlative SyntR**, as in

notwithstanding a widely accepted viewpoint that in the nominative/accusative case of the quantitative NP it is the numeral that controls the noun requiring that it should appear in the genitive.

In certain special cases, where the words written without a space between them are not composite words but in fact represent whims of Russian orthography, other solutions have been proposed and implemented. This relates in particular to units with the numeral non 'half', as in noncmpahu 'half the country', where non is detached from the second part of the unit and linked to it with the help of the quantitate relation, and to units like help e 'there is no place where', which is divided into the negative existential verb he e 'there isn't' and the pronominal adverb 20e 'where'. The two elements are not directly linked in the structure. For details of their syntactic representation, see Aprjesjan-Iomdin 1990 and Apresjan et al. 2010.



'neither he, nor you, nor I'. The purpose of this provision is threefold: reflect the syntactic unity of a correlative conjunction, capture the similarity of coordination expressed by lone and paired conjunctions, and retain the arboreal character of the syntactic structure.

It should be emphasized that in the course of structure generation ETAP does not produce any additional nodes for words physically absent from the sentence.

In particular, no anaphoric pronouns are introduced in sentences like (1) Иван сказал, что устал (lit. Ivan said that was tired, in which some parsers may add the pronoun oh 'he'), no elliptic omissions, as in (2) Я заказал сок, a он пиво 'I ordered a juice and he a beer' are restored⁵. Moreover, the parser does not even generate special nodes for zero forms of the present tense of the verb 6ыт6 '(irrespective of its particular lexical meaning), which are so common in Russian. Accordingly, the constructions like (3) Oh 6ыл счастлив 'he was happy' or (4) Я буду в отпуске 'I will be on leave' receive parses noticeably different from those generated for sentences like (3a) Oh счастлив 'he is happy' or (4a) Я в отпуске 'I am on leave': compare e.g. parses for (3) and (3a) below:

In some of the applications in which the ETAP parser is used, zero copulas are generated at a later stage of sentence processing.

The syntactic tree of the sentence as generated by the ETAP parser is **ordered**: it retains the information on word ordering of the source sentence.

1.2. Major Linguistic Resources Used

ETAP parser makes use of the following two major types of linguistic resources:

• the grammar, which consists of several hundreds of binary syntactic rules, or syntagms⁶, and

⁵ In the SynTagRus treebank (see below) created with the help of the ETAP parser, elliptic omissions are restored manually. E.g. the parse for (2) receives another node for *3ακα3απ*, which is assigned a special feature PHANTOM.

Note that ETAP understands the syntagm in full compliance with the Meaning Text theory, which differs from the notion of a syntagm accepted in traditional linguistics where it is understood, rather, as a group of words or, in phonetics, as a string of words pronounced as a single word.

• the dictionary. ETAP resorts to the so-called combinatorial dictionary that contains rich and diverse information on every lexical entry. Conceptually, the combinatorial dictionary can in fact be considered as a simplified version of the explanatory combinatorial dictionary of the Meaning ⇔ Text theory, the main difference being that the ETAP dictionary has no explicit lexicographic definitions. At the moment, the combinatorial dictionary has 100,000 entries.

1.2.1. An Example of an ETAP Syntagm

The following syntagm, reproduced in Fig.1, is used to generate the predicative link between the verb in the imperative [X] and its subject in the nominative [Y]: this is a construction peculiar for a specific type of Russian conditional sentences like

(5) Приди [X] он [Y] раньше, мы бы успели все обсудить 'If he came earlier (lit. Come; he earlier...) we would have time to discuss all'.

REG:ПРЕДИК.05 ПОДЛЕЖАЩЕЕ В ИМЕНИТЕЛЬНОМ ПАДЕЖЕ

ПРИ СКАЗУЕМОМ,

N:01 ВЫРАЖЕННОМ ИМПЕРАТИВОМ, В УСЛОВНОМ

ПРЕДЛОЖЕНИИ

CHECK

1.1 =(Х,ПОВ,ЕД)

1.2 R-EQU(X,Y,4,ИМ)/LEXR(X,БЫТЬ)& R-EQU(X,Y,4,РОД)&L-EQU(X,*,0,НЕ1)

 $2.1 \text{ PININT}(X,Y,3\Pi T,1)$

3.1 DEP-EQUN(X,Z,ОБСТ,ЛИЧ,ИНФ)

3.2 DOM-LEXR(Z,*,АНАЛИТ,БЫ)

 $3.3 \text{ PININT}(X,Z,3\Pi T,1)/\text{PININT}(Z,X,3\Pi T,1)$

DO

1 SVUZOT:(X,Y,ПРЕДИК)

Fig. 1. A predicative syntagm of ETAP

Syntagms, as all other rules of ETAP, are written in a special formal language for linguistic descriptions, called FORET, based on three-valued first order predicate logic. Somewhat simplifying the picture, we may say that any syntagm consists of two zones: (i) the CHECK zone, which lists the conditions to be verified written with the help of **predicates**, and (ii) the DO zone, which contains an **instruction** to the algorithm to create a hypothetical syntactic link; this instruction is to be performed if all the conditions of the CHECK zone are satisfied.

All conditions are written in the disjunctive normal form and arranged into several groups, identified by the first figure in the two-position number of the condition. Items belonging to the groups where this figure is odd describe **necessary** conditions that have to be satisfied in order for the syntagm to be applied, and those with the even first figure present **impossible** conditions that should **not** be satisfied if the syntagm is to be applied. Obviously, "odd" conditions are, implicitly, conditions with the existential quantifiers, requiring that there should exist at least one variable for which

the condition is satisfied. Conversely, "even" conditions have implicit universal quantifiers: for every variable it should not be true that the condition is satisfied. Further, groups 1 and 2 of the CHECK zone list the conditions that could be checked using only morphological analysis results, the information from the dictionary entries of words present in the sentence processed and the linear order of the words in the sentence. Conditions belonging to groups with larger numbers can only be checked on the ready tree structure, or at least on the fragment thereof as it is generated by the parser.

In the predicative syntagm of Fig. 1, there are conditions of group 1, 2 and 3; hence, all conditions save condition 2.1 are necessary conditions.

Condition 1.1 requires that the sentence processed should have a word which is the imperative (" $\pi o B$ ") in singular (" $e \pi$ "). This is done by the predicate of equation (=). In sentence (5) there is one such word: $npu\partial u$ 'come'.

Condition 1.2 is a disjunction of conjunctions. The predicate in the first elementary disjunction requires that to the right of X, at a distance of no more than 4 words, should be a word Y in the nominative (" μ M"): note the obligatory inversion of the subject! In (5), there are two such words: word #2 (θ H 'he') and word #4 (θ H 'we'). The first disjunction requires that the word X should be the verb θ H be', in which case (second conjunction of this disjunction) to the right of X, again at a distance of no more than 4 words, there should be a word Y in the genitive case (" θ DA"). The last conjunction of this disjunction requires that X be immediately preceded be a negation θ H 'not' (such a situation is presented e. g. by the sentence like θ H by θ H baha, θ C comanoch θ H no-npewhemy 'If Ivan did not exist everything would remain the same'). For sentence (5) this disjunction is irrelevant since it has no occurrence of θ H at all. It is important to understand that the two disjunctions of condition 1.2 are not mutually exclusive: the first disjunction permits that X may be any verb, in the imperative, including θ y θ b, if Y is in the nominative.

The impossible condition 2.1 requires that there should be no comma (" 3π T") between X and Y. For (5), it automatically excludes word #4 from the list of candidates for Y.

Conditions 3.1 to 3.3 introduce the arboreal context of X and Y. Thus, condition 3.1 requires that there should exist a Z which subordinates X with the "обст" (adverbial) SyntR and that this Z should be a fininte verb ("лич") or and infinitive ("инф"). In (5), there are two such words: #6 (успели 'had time') and #8 (обсудить 'discuss').

Condition 3.2 requires that Z should dominate the particle $\delta\omega$ which constitutes the subjunctive mood. This automatically excludes word #8 from the list of candidates for Z.

Finally, condition 3.3 requires that there should be a comma between X and Z: first disjunction relates to the case when Z follows X (as in sentence (5)), and the second disjunction relates to the case when Z precedes X.

It is easy to see that the set of conditions in the CHECK zone of the syntagm are sufficient to identify uniquely the values of all three variables X, Y, and Z for sentence (5): these are respectively, the words $npu\partial u$, oH and ycnenu. Note that the fourth word involved in the rule, $\delta \omega$, is introduced with an anonymous variable (defined by the asterisk in the predicate of condition 3.2). This is possible as $\delta \omega$ does not trigger any other conditions that refer to it.

It is important to understand that the contextual links required in conditions of the third group of the CHECK zone should be established by other syntagms interacting with ours in the parser.

Once the conditions are satisfied, the instruction of the DO zone is performed. In sentence 5, the instruction establishes the hypothetical predicative link ("предик") going from X to Y.

An Example of a Dictionary Entry

Fig. 2 below reproduces a simple combinatorial dictionary entry for the word $npo\partial a wa$ 'sale'. This is in fact only a part of the entry, from which the zone responsible for translation of the word into English is omitted (with the exception of the default translation field in line 24). The Russian language zone proper is reproduced in full.

```
1
   ПРОДАЖА
2.
       POR:S
3
       SYNT:ЖЕНСК.ИСЧИСЛ
4
       DES:'ДЕЙСТВИЕ','ФАКТ','АБСТРАКТ'
5
       D1.1:ТВОР,'ЛИЦО'
6
       D2.1:РОД
7
       D3.1:ДАТ,'ЛИЦО'
       D4.1:3A1, 'ДЕНЬГИ'
8
9
       D4.2:ПО4,НПУСТ,'ДЕНЬГИ'
10
      V0:ПРОДАВАТЬ
11
      SYN1:ТОРГОВЛЯ
12
      CONV:ПОКУПКА
13
      ANTI:ПОКУПКА
14
      S1:ПРОДАВЕЦ
15
      S2:TOBAP
16
      S3:ПОКУПАТЕЛЬ
17
      OPER1:ОСУЩЕСТВЛЯТЬ
18
       OPER2:БЫТЬ<B2>
19
       INCEPOPER2:ПОСТУПАТЬ1<В1>
20 TRAF:AΓEHT.10
21 TRAF:1-КОМПЛ.20
22 TRAF:2-КОМПЛ.21
   **********
23 ZONE:EN
24 TRANS:SALE
```

Fig. 2. A lexical entry of the Russian combinatorial dictionary.

Lines 1–2 indicate the lemma and the part of speech (noun).

Line 3 cites two simple syntactic features that point to the feminine gender of $npo\partial a ma$ and the fact that it is a count noun. These features are used whenever grammatical agreement of the word is to be checked, or verify whether it may form

a quantificative noun phrase. As a matter of fact, the notion of syntactic feature is the most important in ETAP; the system involves over 200 syntactic features, some of them very sophisticated, which determine whether or not the word can be part of a particular syntactic construction.

Line 3 presents semantic features, or descriptors, of the word: in this case 'action', 'fact', and 'abstract'. Descriptors are used to ensure semantic agreement between elements of the sentence processed. Unlike semantic features, the system of descriptors in ETAP is rather simple and straightforward: it includes ca. 40 elements arranged into a weak hierarchy.

Lines 5–9 provide the government pattern of the word. Line 5 says that the first valency (that of the agent, the seller) could be instantiated by a word in the instrumental case that has a semantic descriptor 'person', as in $npo\partial a ma \ Mbahom$ 'sale by Ivan' or $npo\partial a ma \ pupmoù$ 'sale by the firm' ('person', 'unlike 'human', is understood in a broad sense that involves people and organizations of all kinds). Line 6 introduces the object valency that should be in the genitive, as in $npo\partial a ma \ nned a$ 'sale of bread'. Line 7 represents the valency of the addressee (the buyer), which should be filled by a word in the dative with the descriptor 'person', as in $npo\partial a ma \ nod ma$ ('sale to Poland'). Lines 8–9 introduces the valency of price that could be instantiated by the prepositional phrase formed by nned a 'for', or nned a 'as in nned a 'as in nned a 'as in nned a 'for' as in nned a 'for' as in nned a 'as in nned a 'for' as in nned a 'as in nned a 'sale of the descriptor' in the latter case, the phrase conveys the idea of distributiveness and may require additional processing: this is triggered by the special feature Hnned a ("nonempty") of the preposition.

Lines 10 to 19 list values of the different lexical functions (LF) for which $npo\partial a ma$ is the keyword. Of these, lines 10–15 introduce substitute LFs, and lines 17–19 list collocate LFs.

2. Essentials of the Algorithm

2.1. Morphological Analysis as input of ETAP algorithm

During text analysis, the parser proper operates after the morphological analyzer produced a morphological structure (MorphS) for each sentence. MorphS is the ordered sequence of all words of the sentence, each one represented by a lemma name, a POS attribute and a set of morphological features. If a word form is lexically and/or morphologically ambiguous, it appears in the MorphS as a set of objects, somewhat loosely called homonyms, each consisting again of a lemma name, a POS attribute and a set of morphological features. To give an example, sentence

(6) Иностранные рабочие часто плохо знают русский язык (lit. foreign workers often badly know Russian language) 'Foreign workers often have a poor knowledge of Russian'

will receive the MorphS given in Fig. 3:

1	1.1 ИНОСТРАННЫЙ	А,ИМ,МН
2	1.2 ИНОСТРАННЫЙ	А,ВИН,НЕОД,МН
3.	2.1 РАБОЧИЙ1	А,ИМ,МН
4.	2.2 РАБОЧИЙ1	А,ВИН,НЕОД,МН
5	2.3 РАБОЧИЙ2	Ѕ,ИМ,МН,МУЖ,ОД
6	3.1 ЧАСТЫЙ	А,ЕД,СРЕД,КР
7	3.2 YACTO	ADV
8	4.1 ПЛОХОЙ	А,ЕД,СРЕД,КР
9	4.2 ПЛОХО	ADV
10	5.1 ЗНАТЬ1	V,НЕПРОШ,НЕСОВ,МН,З-Л
11	6.1 РУССКИЙ1	А,ИМ,ЕД,МУЖ
12	6.2 РУССКИЙ1	А, ВИН,НЕОД,ЕД,МУЖ
13	6.3 РУССКИЙ2	Ѕ,ИМ,ЕД,МУЖ,ОД
14	7.1 ЯЗЫК1	S,ИМ,ЕД,МУЖ,НЕОД
15	7.2 ЯЗЫК1	S,ВИН.ЕД,МУЖ,НЕОД
16	7.3 ЯЗЫК2	S,ИМ,ЕД,МУЖ,НЕОД
17	7.4 ЯЗЫК2	S,ВИН.ЕД,МУЖ,НЕОД
18	7.5 ЯЗЫКЗ	S,ИМ.ЕД,МУЖ,ОД

Fig. 3. Morphological structure of a sentence

Here, A, ADV, S, and V denote, respectively, the adjective, adverb, noun and verb; ИМ and ВИН stand for the nominative and the accusative; ЕД and МН mark the singular and plural numbers. МУЖ and СРЕД denote the masculine and the neutral gender. КР represents the short form of the adjective. ОД and НЕОД represent the animateness/inanimateness of adjectives and nouns. НЕПРОШ, НЕСОВ and 3-Л show the present tense, the imperfective aspect and the third person of the verb. As it happens, all words of (6) except 5 (the verb 'know') are ambiguous. In particular, word 6 is lexically ambiguous between adjective 'Russian' and noun 'the Russian', both varying in case marking; words 3 and 4 may both be interpreted as adverbs ('often', 'badly') or adjectives ('frequent', 'bad'), whilst word 7 has three lexical readings corresponding to 'language', 'tongue', and 'prisoner', of which the former two, being inanimate, have the same forms for the nominative and the accusative case.

Accordingly, (6) consisting of 7 words has a MorphS that has as many as 18 homonyms.

The morphological analyzer is based on a comprehensive morphological dictionary of Russian that counts over 130,000 entries (over 4 million word forms). ETAP has no separate POS tagger; however, there is a small post-morphological module that partially resolves lexical and morphological ambiguity taking account of near linear context. In the case of sentence (6), this module will only delete 2 homonyms and reduce the strength of one more. On average, the module purges less than 20% of homonyms.

2.2. Creation of the Set of Hypothetical Syntactic Links

ETAP takes a MorphS of a sentence processed as **input** and builds a dependency tree for this sentence using syntagms. At the first stage of the algorithm, the parser constructs all possible hypothetical links, which is performed in a number of steps. The primary list (the so-called matrix of hypotheses) is built exclusively on account of the linear conditions of the syntagms (see above). After that, conditions belonging to groups with higher numbers are applied to the matrix: at this stage, different methods of backtracking are used.

After all conditions of the syntagms have been verified, the algorithm resorts to a number of **filters** aimed at deleting excessive links so that the remaining ones form a dependency tree.

These filters are of diverse nature and may involve

- · data on agreement or government,
- repeatability/non-repeatability of specific syntactic relations (e.g. a verb may have several adverbial modifiers attached by the adverbial relation but only one subject or one direct object attached by the predicative or 1st completive relation⁷),
- data on link projectivity (by default, any link is projective unless a set of specific conditions are met⁸).

Importantly, the parser has three sets of rules in addition to syntagms, resorted to in the process of tree generation. These include, in order of application, 1) intersyntactic rules; 2) top node selection rules and 3) preference rules.

2.3. Intersyntactic Rules

The so-called **intersyntactic** (**INTERSYNT**) rules operate on the whole set of hypothetical rules after all conditions of syntagms have been checked. These rules are designed to **prioritize** the hypotheses produced so that the subsequent stages of the algorithm could first choose the hypotheses with higher priority. The rules assign certain **weights** to syntactic hypotheses as well as to different homonyms of an ambiguous word on the basis of empirically found regularities that involve POS information, type of lexical ambiguity, certain syntactic configurations and the like. At present, this is only done by instructions that increase or reduce the strength of a link or a homonym and do not resort to any numerical values. Accordingly, the newly assigned weights are **absolute** (i. e. we cannot reduce or increase the weight

In case of subject/object coordination, only one predicative or 1st completive relation is established between the predicate and the head of the coordination string (the leftmost member thereof, see above).

It turns out that even though a notable proportion of the links in dependency trees are non-projective (averagely, about 10% of processed sentences contain at least one non-projective link), the share of such links in the total amount of produced links is less than 1%.

of one link or homonym with respect to another concrete link or homonym). Despite this, INTERSYNT shows a rather satisfactory performance, which positively affect the quality of the parser.

Some of the INTERSYNT rules take account of **lexical co-occurrences**. E.g. if a sentence contains a collocation that is likely to be considered as an argument of a lexical function and its value, such a collocation is prioritized: the link that connects the part of such a collocation and/or homonyms that constitute it are assigned high weight values. A useful, if somewhat eccentric, technique applied in INTERSYNT rules is prioritizing links and homonyms of a collocation that are supplied with rules of non-standard translation into English (let it be reminded that originally ETAP was built specifically for machine translation).

An important recent innovation in this mechanism is the creation of INTERSYNT rules that in fact reproduce the most important syntagms, which form the bulk of the syntax, with a vital difference that the syntagms' conditions are formulated for a drastically simplified environment (shorter distances between the head and the daughter, default word order ignoring rarely occurring inversions, prototypical instantiations of variables, e. g. only nouns are included but not their syntactic equivalents like numerals. substantivized adjectives or participles etc.). If in a sentence processed the conditions of such a rule are met, the respected link is assigned a high weight value; respectively, the link generated by the "parent" syntagm is likely to appear in the resulting tree (see in particular Tsinman-Druzhkin 2008).

2.4. Top Node Selection Rules

The so-called **top node selection** rules arrange possible candidates for the absolute head of the future tree structure according to the empirical likelihood principle. Hand-written rules of this ordering take account of a number of different factors (part of speech, morphological features, linear position in the sentence, close environment, presence or absence of hypothetical links going to and from the word tested etc.) and perform fairly well. For example, a finite verb \mathbf{X}_1 is more likely to act as head of the sentence than another finite verb \mathbf{X}_2 located to the right of \mathbf{X}_2 ; however, the situation reverts if \mathbf{X}_1 is preceded by a subordinating conjunction, in which case \mathbf{X}_1 will probably depend on this conjunction in the subordinate clause and \mathbf{X}_2 will be more likely to act as absolute head.

It should be noted that this block of rules is the only one in ETAP when weight values could be relative (there are rules that increase or decrease the weight of some link with regard to another link, whose weight has been established previously).

A recent innovation in this block of rules is the inclusion of **a statistical component**: statistical data are collected from SynTagRus, the Russian treebank created with the help of ETAP.

2.5. General Preference Rules

Preference rules of several types are applied after all intersyntactic rules have been applied and the head is selected. The objective of preference rules is the same as that of INTERSYNT rules: prioritization of the remaining hypotheses. However, preference rules, unlike INTERSYNT rules, are not irreversible and the algorithm may roll back if at a particular step the construction of the structure is blocked.

Most preference rules work with syntactic hypotheses, trying to determine which one of a bunch of hypothetical links going to or from a particular word is the most plausible. Other rules prioritize different homonyms of words; some of them may relate to a concrete word that produces lexical and/or morphological ambiguity; others address typical sorts of ambiguity, e.g. lexical pairs that may be composed of nominal and adjectival lexemes, like *pycckuŭ1* 'Russian' or *pycckuŭ2* 'a Russian', of active or passive verbs, like *oka3ываться*, which may be either the infinitive of an active verb 'turn out' or the passive infinitive of the verb *oka3ывать* 'give, offer'.

If after all these rules have been applied and no tree can be chosen because extra hypotheses are still present, the algorithm resorts to the exhaustion of the remaining alternatives. In the standard situation, the algorithm starts by eliminating one link of the remaining set, finding it in accordance with the preset graph transversal subroutine, uses recursion and rollback mechanisms if needed, until a tree is produced.

Recently, a modified technique of alternatives exhaustion has been introduced, which once again resorts to statistics collected from SynTagRus. This technique uses a greedy algorithm of choosing the links to be deleted based on the evaluation of probabilities of their correctness. To collect evaluation data, the parser is run on SynTagRus sentences, in which for every pair of alternating hypotheses we know which of them is correct, or know that both are incorrect. Pairwise probabilities are used to assess the correctness probabilities of for every link belonging to bunches of links entering a word. The evaluation only taken account of names and lenghths of the links and is therefore rather rough but has proven to be fairly efficient.

2.6. Patterns of ETAP Operation

ETAP parser has three patterns of operation, which are called **rapid syntax**, **full syntax**, and **emergency syntax**. The first pattern may be started after INTERSYNT rules have been applied: the algorithm temporarily deletes all weak links and homonyms and strives to build the tree from strong and normal elements alone. If this pattern fails, the algorithm restores the weak links and resumes the work with the whole set of hypothesis: this is the full syntax pattern. Should this pattern fail, too, the algorithm resorts to the emergency syntax pattern, which starts by detecting the node or nodes left without the head and attaching it to some other nodes with the help of a fictitious syntactic link or links, using the so-called soft-fail mechanism. If emergency syntax is activated, the resulting tree may prove to be far from satisfactory.

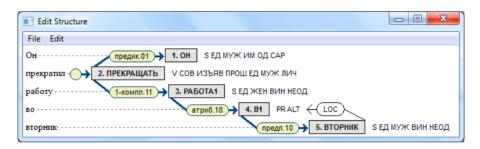
ETAP options allow the user to skip either the rapid syntax or the full syntax pattern, but not both.

3. Major Applications

3.1. ETAP-3 Machine Translation System

Originally, ETAP parser of Russian was intended for machine translation and built specifically for this purpose. Together with the parser for English it constituted the main computational linguistics resource on which the system is based.

This objective naturally determined many of the properties of the parser and concrete solutions taken therein; in particular, the developers placed a very strong emphasis on the lexical aspect of the system, primarily striving to represent in the most precise manner all links that were responsible for the instantiation of valencies of the predicates, while the achievement of overall syntactic accuracy was given a somewhat lesser priority. For example, the parser does not always ensures correct attachment of prepositional phrases, so that the parse for the sentence *Oh прекратил работу во вторник* 'He stopped the work on Tuesday' will attach the temporal modifier *во вторник* 'on Tuesday' to the noun rather than to the verb:



This is acceptable in machine translation tasks as wrong PP-attachment does not normally affect the translation adequacy.

In some cases, decisions were taken to deliberately disregard certain linguistic phenomena in order to simplify the rules. For example, the parser does not build non-projective attributive and adverbial links, although actant links like predicative and 1st completive may well be non-projective.

3.2. SynTagRus Treebank of Russian

Another important application of ETAP is the creation of the first syntactically tagged corpus of Russian, SynTagRus (see e.g. Apresjan et al. 2005)⁹. The corpus is built semiautomatically: for every sentence of a text belonging to the corpus ETAP first builds a syntactic tree, which is then manually checked by at least two human

⁹ SynTagRus is accessible online on the website of the Russian National Corpus (www.ruscorpora.ru) as its subcorpus.

experts, which ensures high quality of the corpus. Human work is facilitated by a powerful software environment, called Structure Editor, which provides a variety of aids to make the process of corpus editing effective and minimize the number of errors (Iomdin-Sizov 2009). It may happen that ETAP cannot at all build a syntactic tree for the sentence (e. g. if it contains an ellipsis); in this case the expert constructs the tree manually, introducing phantom nodes as needed.

At present, the corpus counts a little over 50,000 sentences (over 460,000 words). Despite this relatively limited size, the corpus proves to be extremely useful not only as a linguistic resource but also as a computational resource which can be utilized to collect various statistical data, create training sets for machine learning, and develop automatic parsers (see Nivre-Bogusalvsky-Iomdin 2008). One of the new features of SynTagRus is that it provides, in addition to syntactic annotation, also annotation with collocate lexical functions.

Importantly, SynTagRus is now effectively used by the ETAP parser itself. There are three main uses of the corpus.

First, it provides the statistics of occurrence of the different syntactic constructions, lexical co-occurrences, patterns of ambiguities etc., which is used in several points of the algorithm if the statistical component is activated.

Second, it serves as an efficient and rather accurate evaluation resource, which is used to evaluate the performance of ETAP parser in many respects and so find and resolve some of the system's bottlenecks (see Boguslavsky et al. 2011).

Finally, it is used for regression testing of ETAP. Periodically, ETAP is run on the whole material of the corpus. Sentences that receive parses exactly equivalent to those stored in the corpus (this subset constitutes between 30 and 35 percent of the bulk of the corpus) are selected as basis for regression testing. ETAP is then regularly run on this test set to see if any of the changes introduced in the dictionary, rules, or software mechanisms affected the state of the test set. Regression testing has proven extremely helpful in ensuring the stability of the parser and eventually improving it in many respects. Last but not least, regression testing helps improve the SynTagRus itself: it happens fairly often that discrepancies in parses detected by regression test runs point to erroneous annotation in the corpus, which is then corrected.

3.3. A Hybrid System of Russian Speech Synthesis

ETAP parser has been effectively used in creating a new system of Russian speech synthesis, ETAP-Multiphone (see Iomdin-Lobanov 2009, Iomdin-Lobanov-Getsevich 2011). The idea is that prior to sending the text to the regular synthetic block it is parsed by ETAP supplemented with rules that find prosodically salient elements in the syntactic structure. The elements receive special treatment in the regular synthetic block, which noticeably improves the result of speech generation. Within this project, the morphological dictionary of ETAP was supplemented with information on the phonetic stress of every word form, which naturally included correct rendering of the Russian letter \ddot{e} . This helped improve the performance of ETAP in sentences where words that may be written with \ddot{e} are indeed written in this way.

3.4. A Semantic Analyzer of Text Involving an Ontology

ETAP parser is used in all new systems that are based on, or constitute a part of, the ETAP-3 linguistic processor. One such system is the semantic analyzer of Russian texts that makes use of a specially designed ontology (see Boguslavsky et al. 2010). The new system requires that the parser performs as accurately as possible. Among other things, the parser must ensure that arguments and values of lexical functions occurring in the text processed could be identified correctly. This provides additional incentives for ETAP development.

4. Unsolved Problems and Future Development

To conclude the description of ETAP we will briefly outline the challenges that the system is still facing. The most important challenge is that, so far, the system is not sufficiently robust. In certain cases, the parser fails to produce an adequate or even an acceptable tree structure. This maybe due to a variety of reasons.

The first reason is that the system cannot work reliably on very long sentences (60 words or more) due to the combinatorial explosion and the fact that it has no good heuristic mechanisms of splitting such sentences into linguistically acceptable chunks.

The second reason is that ETAP lacks sufficient external resources, like a named entity recognition component, POS tagger, or a reliable morphological guesser, which reduces its potential of correctly handling sentences with unknown words.

In some cases, linguistic support of ETAP has obvious gaps. In particular, this is manifested in the fact that linguistic rules are sometimes too rigid and are unable to cope with sentences that contain deviations of the prescribed standard (metaphorical uses of words, irregular instantiation of valencies and the like); besides, it has no proper mechanisms of handling elliptical sentences of many kinds.

Additionally, ETAP has certain inadequacies in the core algorithm. In particular, soft-fail mechanisms that are used in the emergency syntax pattern of operation are rather rough and, instead of providing a structure with only local defects, may sometimes play havoc with the result.

All these challenges are now being addressed. The developers of ETAP are working to create the necessary resources, including the POS tagger and the morphological guesser, partially using machine learning techniques. Special efforts are also made to convert the parser into a hybrid system that combines rule-based and machine learning approaches.

References

- 1. *Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin* et al. (1989). Lingvisticheskoe obespechenie sistemy ETAP-2 [The linguistics of the ETAP-2 MT system]. Moscow, Nauka. 295 p.
- 2. *Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin* et al. (1992). Lingvisticheskij protsessor dlja slozhnyx informatsionnyx sistem [A linguistic processor for advanced information systems]. Moscow, Nauka, 1992. 256 p.
- 3. Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin, Alexandre Lazourski, Vladimir Sannikov, Victor Sizov, Leonid Tsinman. (2003). ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT, MTT 2003, First International Conference on Meaning Text Theory (June 16–18 2003). Paris: Ecole Normale Supérieure, P. 279–288.
- Jurij Apresjan, Igor Boguslavsky, Leonid Iomdin, Vladimir Sannikov (2010). Teoreticheskie problemy russkogo sintaksisa. Vzaimodejstvie grammatiki I slovarja [Theoretical Issues of Russian Syntax. Interaction of the Grammar and the Dictionary]. Moscow. Yazyki Slavyanskix kultur publishers. ISBN 978-5-9551-0386-0. 408 p.
- 5. *Jurij Apresjan, Leonid Iomdin* (1990). Konstruktsii tipa NEGDE SPAT' v russkom jazyke: sintaksis i semantika [Constructions of the NEGDE SPAT' type in Russian: Syntax and semantics.], Semiotika i informatika [Semiotics and Informatics], No. 29. Moskva, 1990, pp. 3–89.
- Jurij Apresian, Leonid Iomdin, Boris Iomdin et al. (2005). Sintaksicheski i semanticheski annotirovannyj korpus russkogo jazyka (sovremennoe sostojanie I perspektivy [Syntactically and Semantically Annotated Corpus of Russian: State-of-the-Art and Prospects] // Natsionalnyj korpus russkogo jazyka 2003–2005 g. (rezul'taty i perspektivy). [National Corpus of Russian 2003–2005 (Results and Prospects)]. Moscow, Indrik. P.193–214.
- 7. *Igor Boguslavsky, Leonid Iomdin, Victor Sizov, Svetlana Timoshenko* (2010). Interfacing the Lexicon and the Ontology in a Semantic Analyzer, COLING 2010. Proceedings of the 6th Workshop on Ontologies and Lexical Resources (Ontolex 2010). Beijing, August 2010. P. 67–76.
- 8. *Igor Boguslavsky, Leonid Iomdin, Leonid Tsinman, Victor Sizov, and Vadim Petrochenkov* (2011). Rule-Based Dependency Parser Refined by Empirical and Corpus Statistics, International Conference on Dependency Linguistics. exploring dependency grammar, semantics, and the lexicon. Kim Gerdes, Eva Hajicova, Leo Wanner (eds). Depling 2011, Barcelona, September 5–7 2011. ISBN 978-84-615-1834-0. P. 318–327. http://depling.org/proceedingsDepling2011.
- 9. Igor Boguslavsky, Leonid Iomdin, Victor Sizov, Denis Valeev. (2008). Sintaksicheskij analizator sistemy ETAP i ego otsenka s pomoshchju gluboko razmechennogo korpusa russkix tekstov. [The syntactic analyzer of the ETAP system and its evaluation with the help of a deeply annotated corpus of Russian texts]. Труды Международной конференции "Корпусная лингвистика -2008" [Corpus Linguistic 2008. International Conference]. Saint Petersburg, Saint Petersburg State University. ISBN 978-5-288-04769-5. p. 56–74.

- 10. Leonid Iomdin, Boris Lobanov (2009). Sintaksicheskie korreljaty prosodicheski markirovannyh elementov predlozhenija [Syntactic Correlates of Prosodically Marked Sentence Elements], Dialog 2009. Kompjuternaja lingvistika i intellektual'nye texnologii. Po materialam ezhegodnoj Mezhdunarodnoj konferentsii "Dialog" (Bekasovo, 27–31 maja 2009 g.). [Dialog 2009. Computational Linguistics and Intellectual Technologies. International Conference]. Moscow, RGGU Publishers, 2009. Issue 8(15). ISBN 978-5-7281-1102-3. P. 136–142.
- 11. Leonid Iomdin, Boris Lobanov, Yuri Getsevich (2011). Govorjashchij ETAP. Opyt ispol'zovanija sintaksicheskogo analizatora sistemy ETAP v russkom rechevom sinteze. [Talking ETAP. Using the Syntactic Analyzer of the ETAP System in Russian Speech Synthesis], Kompjuternaja lingvistika i intellektual'nye texnologii. Po materialam ezhegodnoja Mezhdunarodnoj konferentsii "Dialog" (Bekasovo, 25–29 maja 2011 g.) [Dialog 2011. Computational Linguistics and Intellectual Technologies. International Conference]. Moscow, RGGU Publishers, Issue 10(17). ISSN 2221-7932. P. 269–279.
- 12. *Leonid Iomdin, Victor Sizov* (2009). Structure Editor: a Powerful Environment for Tagged Corpora, MONDILEX Fifth Open Workshop, Ljubljana, Slovenia, 14–15 October, 2009. Ljubljana. ISBN 978-961-264-012-5. P. 1–12.
- 13. *Igor Mel'čuk* (1974/1999). Opyt teorii lingvisticheskih modelej Smysl ⇔ Tekst. [The theory of linguistic models "Meaning ⇔ Text']. Moscow, Nauka; Jazyki russkoj kultury.
- Joakim Nivre, Igor Boguslavsky, Leonid Iomdin (2008). Parsing the SYNTAGRUS Treebank of Russian, Coling 2008. 22nd International Conference on Computational Linguistics. Proceedings of the Conference. Vol. 2. ISBN: 978-1-905593-47-7. P. 641–648.
- 15. Leonid Tsinman, Konstantin Druzhkin (2008). Sintaksicheskij analizator lingvisticheskogo protsessora ETAP-3: Èksperimenty po ranzhirovaniju sintaksicheskih gipotez [The syntactic analyzer of the ETAP-3: experiments on prioritizing syntactic hypotheses], Dialog 2008. Kompjuternaja lingvistika i intellektual'nye texnologii. Po materialam ezhegodnoja Mezhdunarodnoj konferentsii "Dialog" (Bekasovo, 4–8 ijunja 2008 r. Computational Linguistics and Intellectual Technologies. International Conference]. Moscow, RGGU Publishers, Issue 7(14). P. 147–153. ISBN 978-5-7281-1022-4.