Exploiting the Russian National Corpus in the Development of a Russian Resource Grammar

Tania Avgustinova
DFKI GmbH & Saarland University
P.O. Box 151150
66041 Saarbrücken, Germany
avgustinova@dfki.de

Yi Zhang
DFKI GmbH & Saarland University
P.O. Box 151150
66041 Saarbrücken, Germany
yzhang@coli.uni-sb.de

Abstract

In this paper we present the on-going grammar engineering project in our group for developing in parallel resource precision grammars for Slavic languages. The project utilizes DELPH-IN software (LKB/[incr tsdb()]) as the grammar development platform, and has strong affinity to the LinGO Grammar Matrix project. It is innovative in that we focus on a closed set of related but extremely diverse languages. The goal is to encode mutually interoperable analyses of a wide variety of linguistic phenomena, taking into account eminent typological commonalities and systematic differences. As one major objective of the project, we aim to develop a core Slavic grammar whose components can be commonly shared among the set of languages, and facilitate new grammar development. As a showcase, we discuss a small HPSG grammar for Russian. The interesting bit of this grammar is that the development is assisted by interfacing with existing corpora and processing tools for the language, which saves significant amount of engineering effort.

Keywords

Parallel grammar engineering, corpora, Slavic languages

1. Introduction

Our long-term goal is to develop grammatical resources for Slavic languages and to make them freely available for the purposes of research, teaching and natural language applications. As one major objective of the project, we aim to develop and implement a core Slavic grammar whose components can be commonly shared among the set of languages, and facilitate new grammar development. A decision on the proper set up along with a commitment to a reliable infrastructure right from the beginning are essential for such an endeavor because the implementation of linguistically-informed grammars for natural languages draws on a combination of engineering skills, sound grammatical theory, and software development tools.

1.1 DELPH-IN initiative

Current international collaborative efforts on deep linguistic processing with Head-driven Phrase Structure Grammar [1-3] exploit the notion of shared grammar for the rapid development of grammars for new languages and for the systematic adaptation of grammars to variants of the same language. This international partnership, which became popular under the name DELPH-IN[1], is based on a shared commitment to re-usable, multi-purpose resources and active exchange. Its leading idea is to combine linguistic and statistical processing methods for getting at the meaning of texts and utterances. Based on contributions from several member institutions and joint development over many years, an open-source repository of software and linguistic resources has been created that already enjoys wide usage in education, research, and application building.

In accord with the DELPH-IN community we view rule-based precision grammars as linguistically-motivated resources designed to model human languages as accurately as possible. Unlike statistical grammars, these systems are hand-built by grammar engineers, taking into account the engineer's theory and analysis for how to best represent various syntactic and semantic phenomena in the language of interest. A side effect of this, however, is that such grammars tend to be substantially different from each other, with no best practices or common representations.[2]

As implementations evolved for several languages within the same common formalism, it became clear that homogeneity among existing grammars could be increased and development cost for new grammars greatly reduced by compiling an inventory of cross-linguistically valid (or at least useful) types and constructions. To speed up and simplify the grammar development as well as provide a common framework, making the resulting grammars more

---

[1] Deep Linguistic Processing with HPSG Initiative (DELPH-IN), URL: http://www.delph-in.net/

[2] Exceptions do exist, of course: ParGram (Parallel Grammar) project is one example of multiple grammars developed using a common standard. It aims at producing wide coverage grammars for a wide variety of languages. These are written collaboratively within the linguistic framework of Lexical Functional Grammar (LFG) and with a commonly-agreed-upon set of grammatical features.

URL: http://www2.parc.com/isl/groups/nltt/pargram/

comparable the LinGO[3] Grammar Matrix[4] has been set up as a multi-lingual grammar engineering project [4] which provides a web-based tool designed to support the creation of linguistically-motivated grammatical resources in the framework of HPSG [5].

The Grammar Matrix is written in the TDL (type description language) formalism, which is interpreted by the LKB[5] grammar development environment [6]. It is compatible with the broader range of DELPH-IN tools, e.g., for machine translation [7], treebanking [8] and parse selection [9].

1.2 LinGO Grammar Matrix

Generally speaking, the Grammar Matrix is an attempt to distill the wisdom of already existing broad coverage grammars and document it in a form that can be used as the basis for new grammars. The main goals are to develop in detail semantic representations and the syntax-semantics interface, consistent with other work in HPSG; to represent generalizations across linguistic objects and across languages; and to allow for very quick start-up as the Matrix is applied to new languages.

The fact that different parts of a single grammar can be abstracted into separate, independent modules, either for processing or grammar development, is approached in [10] from the perspective of reuse of grammar code. A web-based configuration system elicits typological information from the user-linguist through a questionnaire [10, 11] and then outputs a grammar consisting of the Matrix core plus selected types, rules and constraints from the libraries according to the specifications in the questionnaire, and lexical entries for the language in question. In other words, users specify phenomena relevant to their particular language, with their selections being compiled from libraries of available analyses into a starter grammar which can be immediately loaded into the LKB grammar development environment [6], as well as the PET parser [12], in order to parse sentences using the rules and constraints defined therein. The regression testing facilities of [incr tsdb()] allow for rapid experimentation with alternative analyses as new phenomena are brought into the grammar [13]. The original Grammar Matrix consisted of types defining the basic feature geometry, e.g. [14], types for lexical and syntactic rules encoding the ways that heads combine with arguments and adjuncts, and configuration files for the LKB grammar development environment [6] and the PET system [12]. Subsequent releases have refined the original types and developed a lexical hierarchy, including linking types for relating syntactic to semantic arguments, and the constraints required to compositionally build up semantic representations in the format of Minimal Recursion Semantics [15-17]. The constraints in this 'core' Matrix are intended to be language-independent and monotonically extensible in any given grammar. In its recent development, the Grammar Matrix project aims at employing typologically motivated, customizable extensions to a language-independent core grammar.

The implemented prototype consists of a small set of modules targeting basic word order (addressing the relative order of subjects, verbs, and verbal complements), sentential negation, main-clause yes-no questions, and a small range of lexical entries. In particular:

- The Matrix core grammar provides definitions of basic head-complement and head-subject schemata which are consistent with the implementation of compositional semantics [16], as well as definitions of head-initial and head-final phrase types. The **word order** module creates subtypes joining the head-complement and head-subject schemata with the types specifying head/dependent order, creates instances of those types as required by the LKB parser, and constrains the rules to eliminate spurious ambiguity in the case of free word order.
- For **yes-no questions**, four alternatives have been implemented: inversion of the subject and a main or auxiliary verb relative to declarative word order and sentence initial or final question particles.
- The **sentential negation** module handles two general negation strategies: via verbal inflection or via a negative adverb. Neither, either or both of these strategies may be selected.
- In a strongly lexicalist theory like HPSG, words tend to carry quite a bit of information, which is reflected in the **lexicon** structure. This information is encoded in lexical types; lexical entries merely specify the type they instantiate, their orthographic form, and their semantic predicate. Many of the constraints required (e.g., for the linking of syntactic to semantic arguments) are already provided by the core Matrix. However, there also is cross-linguistic variation. The forms are assumed to be fully inflected (modulo negation), support morphological processes awaiting future work. This information and the knowledge

[3] The Linguistic Grammars Online (LinGO) team is committed to the development of linguistically precise grammars based on the HPSG framework, and general-purpose tools for use in grammar engineering, profiling, parsing and generation. URL: http://lingo.stanford.edu/

[4] URL: http://www.delph-in.net/matrix/

[5] LKB (Linguistic Knowledge Builder) system is a grammar and lexicon development environment for use with unification-based linguistic formalisms. While not restricted to HPSG, the LKB implements the DELPH-IN reference formalism of typed feature structures (jointly with other DELPH-IN software using the same formalism).

base are used to produce a set of lexical types inheriting from the types defined in the core Matrix and specifying appropriate language-specific constraints, and a set of lexical entries.

In a lexicalist constraint-based framework, the grammars are expressed as a collection of typed feature structures which are arranged into a hierarchy such that information shared across multiple lexical entries or construction types is represented only on a single supertype. As a result, a cross-linguistic type hierarchy comes with a collection of phenomenon-specific libraries.

2. Typologically motivated modularity

Aiming at typologically motivated modularity [10] describe a method for extending a language-independent core grammar with modules handling cross-linguistically variable but still recurring patterns. This method allows for extremely rapid prototyping of HPSG-conform grammars in such a way that the prototypes themselves can serve as the basis for sustained development, being able to scale up to broad-coverage resource grammars. The authors envision four potential uses for such grammar prototyping: (i) in pedagogical contexts, where it would allow grammar engineering students to more quickly work on cutting-edge problems; (ii) in language documentation, where a documentary linguist in the field might be collaborating remotely with a grammar engineer to propose and test hypotheses; (iii) in leveraging the results from economically powerful languages to reduce the cost of creating resources for minority languages; and (iv) in supporting typological or comparative studies of linguistic phenomena or interactions between phenomena across languages.

The modular approach of [10] has been designed to handle two kinds of typological variation. On the one hand, there are systems (formal or functional) which must be represented in every language. For example, every language has some set of permissible word orders (formal) and a means of expressing sentential negation (functional). On the other hand, there are linguistic phenomena which appear in only some languages, and are not typically conceptualized as alternative realizations of some universal function, phenomena such as noun incorporation, numeral classifiers, and auxiliary verbs. It is indeed expected that the constraint definitions which are supplied to grammar developers can be extended to capture generalizations holding only for subsets of languages.

The strategy of [10] is actually consistent with data driven investigation of linguistic universals and constraints on cross-linguistic variation. Therefore, we refer to this approach of grammar development (with the help from Grammar Matrix) as the "bottom-up" approach, because it is driven purely by the specific phenomena in the target language. It is certainly efficient: Specifying the choice file and building a small working grammar can be done within an hour (excluding the time spent on deciding specific choices for the given language). For instance [18] reports a relatively short development time required to create a precision, hand-built grammar for the Australian language Wambaya as a qualitative evaluation of the Grammar Matrix as a cross-linguistic resource.[6] The major drawback of this approach, however, is that, for the set of customized grammars for a group of languages, it soon becomes difficult (if not impossible) to harmonize the treatment of related phenomena across languages. With grammars being created individually, the treatment of shared phenomena would work to the degree that satisfies but does not guarantee cross-linguistic compatibility. As the number and breadth of implemented grammars grows, linguistic predictions are expected to emerge and become part of improved modules, particularly with respect to interactions among the distinct phenomena covered. Our focus in creating a Slavic Grammar Matrix is therefore somewhat different: since we are dealing with representatives of a language family, this effectively enables a “top-down” perspective in the multilingual grammar design.

It is an appealing goal indeed to develop a theoretical account of the way that language variation may be described in HPSG. Crucial in this respect is the fact that the HPSG framework allows a clean way of encoding at least some aspects of language variability within the type system. Another more ambitious line of research is initiated for investigating whether the description of differences among any set of two or more languages can be reduced to a minimal set of types. Progress in this research will lead to shared portions of grammar since the similarity of phenomena among the different languages will then be reflected in identical HPSG descriptions within the type systems. The goal is a grammar matrix designed for maximum reusability, lifting out the elements that can and should be common across HPSG grammars. Therefore, it is important to determine which analyses or building blocks of analyses appear to be cross-linguistically applicable. As the grammar matrix won’t be a complete grammar fragment by itself, it will be used in combination with mini-grammars for various languages. For a language family, and closely related languages in general, it is certainly justified to introduce intermediate

[6] Despite large typological differences between Wambaya and the languages on which the development of the resource was based, the Grammar Matrix is found to provide a significant jump-start as the creation of grammar itself is reported to have taken less than 5.5 person-weeks of effort.

parameterizations of the cross-linguistic core of the grammar matrix.

2.1 SlaviGraM: Slavic Grammar Matrix

The common properties of Slavic languages have been observed both in literature and related research at various intermediate levels of linguistic abstraction. Intermediate levels of typological variation are essential to our project because we work with a closed set of well-studied, well-documented and generally resource-rich languages belonging to the same language family. In this context, the interesting question arises whether minimal differences are also detectable as parameters of systematic variation.

Our concept of Slavic core grammar (Figure 1) will shape up and crystallize through rigorous testing in parallel grammar engineering for a closed set of richly documented and well studied genetically related languages for which a variety of linguistic resources is already available. We use Grammar Matrix to quickly build small grammars for individual languages, utilizing the online Matrix configuration system[7] to specify choice files for representatives of each Slavic subgroup, namely for Russian (East Slavic), Bulgarian (South Slavic) and Polish (West Slavic), as an initial step.

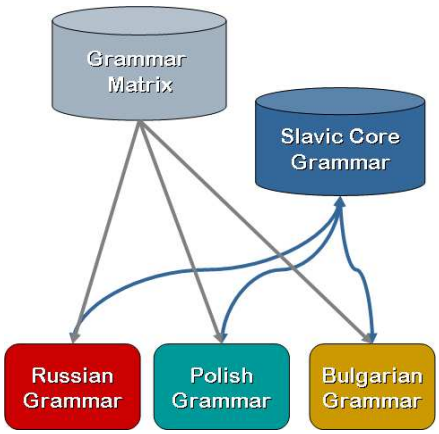
Apart from the shared core in the Grammar Matrix, however, the customization script treats the individual languages as separate instances, which means that the fact that we have to do with a group of closely related languages cannot be taken into account in the original setting. Therefore, shared analyses from individual languages are put into the Slavic Core in the form of generalized Slavic hierarchy and libraries. When new language is added, the Slavic core helps to more efficiently build the new grammar, and potentially receives cross-Slavic validation.

[7] The system consists of the following three parts:

- **Customization Page**. In order for the system to create a starter grammar, the required information must be elicited from the user-linguist. The medium for this elicitation is a web interface.
- **Choices File**. The options selected by the user are saved in a plain text file, called the choices file. Before a grammar is built, the choices file is verified to be internally consistent and contain all the information it needs.
- **Customization Script**. Matrix grammars are written in a type description language (TDL). The customization script is a Python script that reads in the choices file, and uses the information it contains to select or construct relevant sections of TDL code. The output is a collection of files containing the language-specific TDL code. This is then bundled with the core Matrix files to provide a small but functioning grammar fragment.

Our approach to Slavic grammatical resources is unique in the sense that grammar engineering for each individual language takes place in a common Slavic setting. This in particular means that if for example two possibilities are conceivable of how to model a particular phenomenon observed in a certain Slavic language, then we strongly prefer the option that would potentially be consistent with what is found in the other grammars. As a result the Matrix-driven starter grammars for Russian and Bulgarian, the two typological extremes within the Slavic language family, eventually incorporate novel theoretical decisions even for seemingly trivial tasks.

Figure 1: Matrix-driven starter grammars in Slavic core grammar setting



The Grammar Matrix in combination with the Slavic Core Grammar allows new grammars to directly leverage the expertise in grammar engineering gained in extensive work on previous grammars of the same language family. Both the general LinGO grammar Matrix and the Slavic Core Grammar are not static objects, but are designed to evolve and be refined as more languages are covered. The advantage of separating the Slavic Core Grammar from the general grammar Matrix is that the closed set of languages under consideration allow our Slavic Core to evolve more liberally than the grammar Matrix, without concerns over unstudied languages.

3. Focus on Russian Resource Grammar

As a showcase, let us consider the he Russian HPSG grammar, which is currently under active construction in our group. In fact, the Russian Resource Grammar has a central position in the SlaviGraM project and is anticipated as a major outcome in terms of end product and a large-scale experimental set up for hypothesis testing. The interesting aspect of the initial Russian grammar is that its development is assisted by interfacing with existing corpora and processing tools for the

language, which saves significant amount of engineering effort.

3.1 Morphological pre-processing

The morphological information associated with word forms in the disambiguated part of the Russian National Corpus, i.e. where the full analysis is displayed, is structured into four fields:

(i) lexeme and its part of speech;

(ii) (word-classifying invariable features (for example, gender for nouns and transitivity for verbs);

(iii) word-form specific inflectional features (for example, case for nouns and number for verbs);

(iv) non-standard forms, orthographic variations, etc. In the rest of the corpus only the lexeme and the part of speech are displayed.

Morphological analysis is the basic enabling technology for many kinds of text processing. Integrating a morphological analyzer is a crucial prerequisite for all grammar development activities involving Slavic languages. For research purposes, such systems are by and large freely available nowadays, and the LKB grammar engineering environment provides the required interface for integrating a morphological pre-processor.

For the pre-processing module we have considered two morphological analyzers for Russian: Mystem [19] and Dialing [20]. Both systems of are based on finite state transducers and have been used in the Russian National Corpus. Unlike Mystem, the system Dialing covers both inflectional and derivational morphology and is based on a large dictionary which also contains information on inflections, prefixes and affixes and stress patterns. Mystem, however, is the morphological component used by the popular Russian search engine Yandex. The underlying algorithm for analysis and synthesis achieves quite precise inflectional morphology of a wide lexical coverage without implying any particular morphological model of the dictionary.

The fact that Mystem is available for Polish too is an additional criterion in favor of adopting it in our project. Thus, during the preparatory phase, we have chosen the system Mystem, and it is already integrated as a morphological pre-processor in the LKB environment, as illustrated in Figure 2.

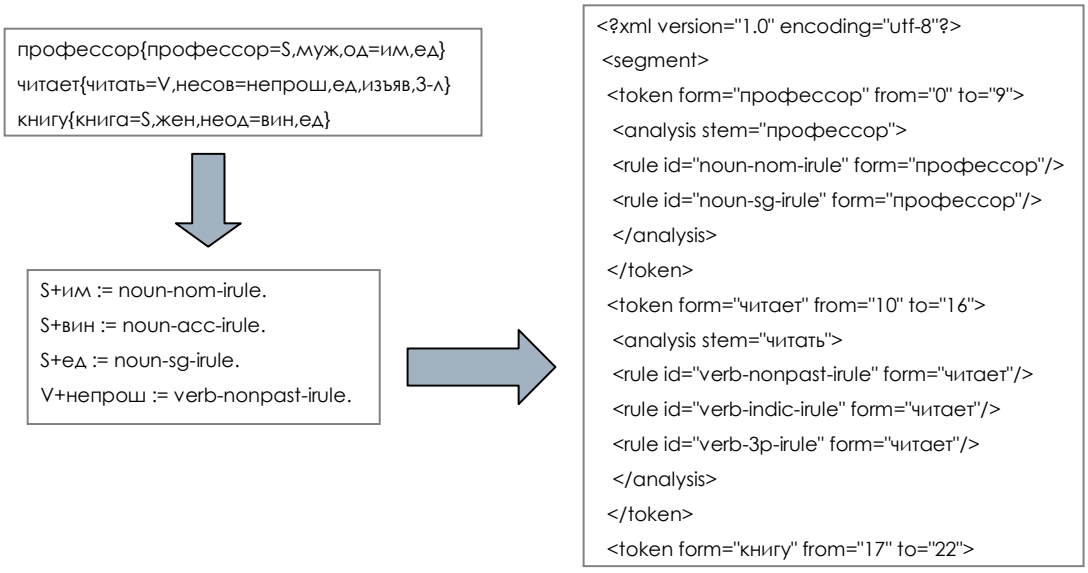
The Russian National Corpus is without a doubt an important source of structured grammatical knowledge to be integrated in our Russian grammar. A snapshot of the main search interface to the RNC is given in Figure 3, while Figure 6 illustrates the access to the syntactically annotated and disambiguated sub-corpus of the RNC. Furthermore, Figure 4 gives us the inventory of morphologically relevant “grammatical features” to select from in the main corpus. Note, however, that the inventory of grammatical features accessed from the syntactic search page is somewhat different, as shown by Figure 5.

Unlike the morphologically annotated portion of the RNC, the deeply annotated sub-corpus only contains fully disambiguated annotations (i.e. both morphological and syntactic ambiguity is resolved).

3.2 Syntactic dependencies

In the deeply annotated sub-corpus of the RNC, every sentence is marked up with a dependency syntactic structure – cf. Figure 7, with nodes corresponding to the words of the sentence, and labeled edges encoding the syntactic relations.

Figure 2: Morphological input via inflectional rules


профессор{профессор=S,муж,од=им,ед}
читает{читать=V,несов=непрош,ед,изъяв,3-л}
книгу{книга=S,жен,неод=вин,ед}
S+им := noun-nom-irule.
S+вин := noun-acc-irule.
S+ед := noun-sg-irule.
V+непрош := verb-nonpast-irule.
<?xml version="1.0" encoding="utf-8"?>

<token form="профессор" from="0" to="9">
<analysis stem="профессор">
<rule id="noun-nom-irule" form="профессор"/>
<rule id="noun-sg-irule" form="профессор"/>
</analysis>
</token>
<token form="читает" from="10" to="16">
<analysis stem="читать">
<rule id="verb-nonpast-irule" form="читает"/>
<rule id="verb-indic-irule" form="читает"/>
<rule id="verb-3p-irule" form="читает"/>
</analysis>
</token>
<token form="книгу" from="17" to="22">

Figure 3: RNC search

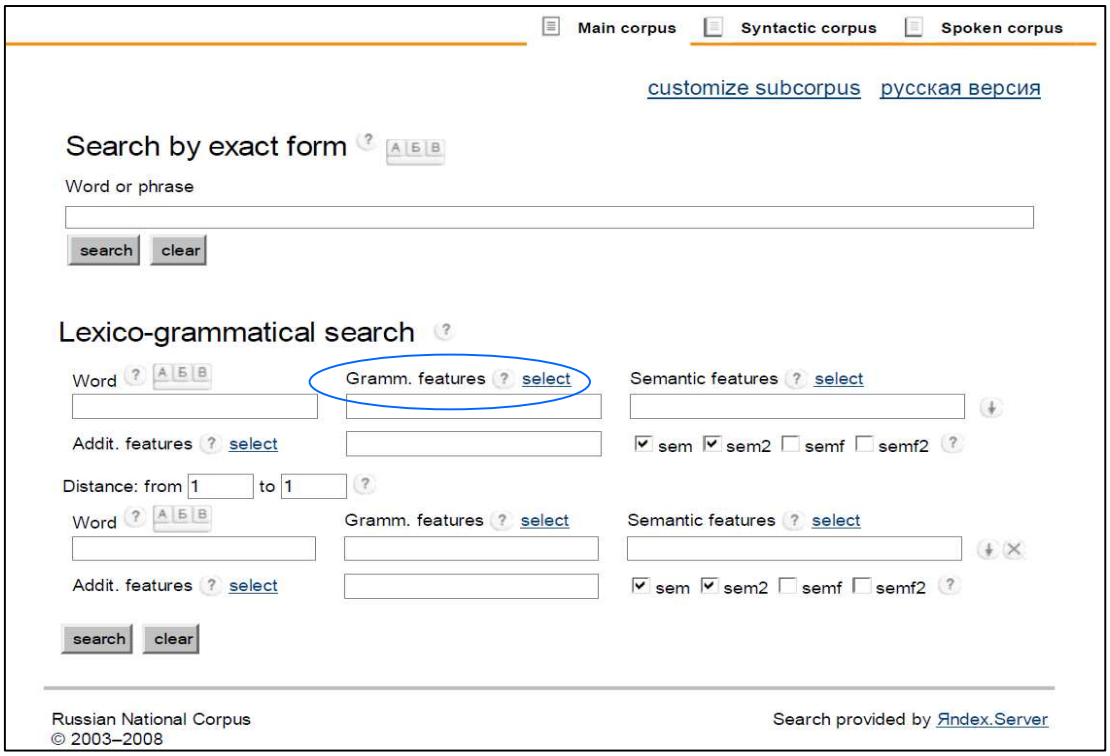


Figure 4: Morphological information in RNC

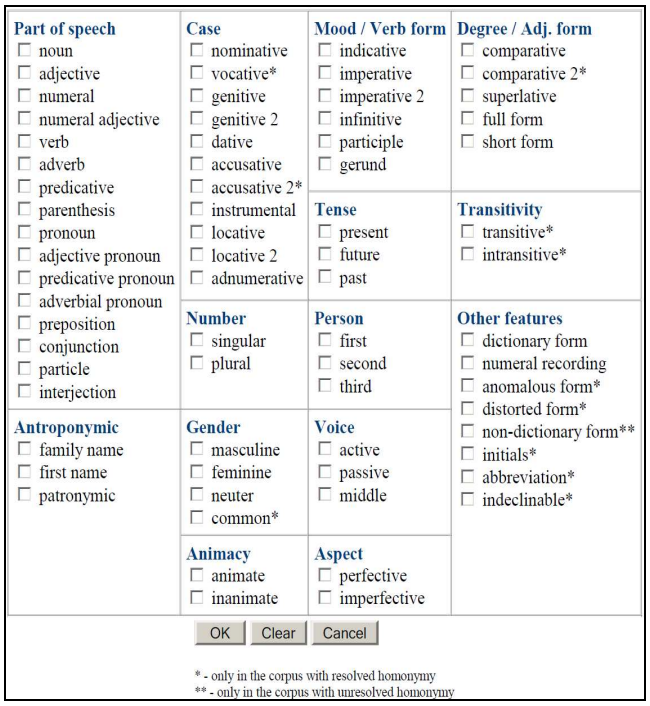


Figure 5: Grammatical features used in the syntactic sub-corpus

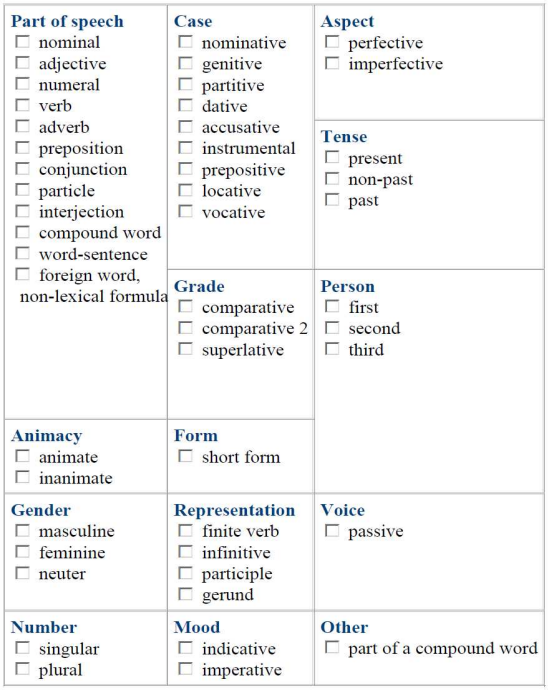
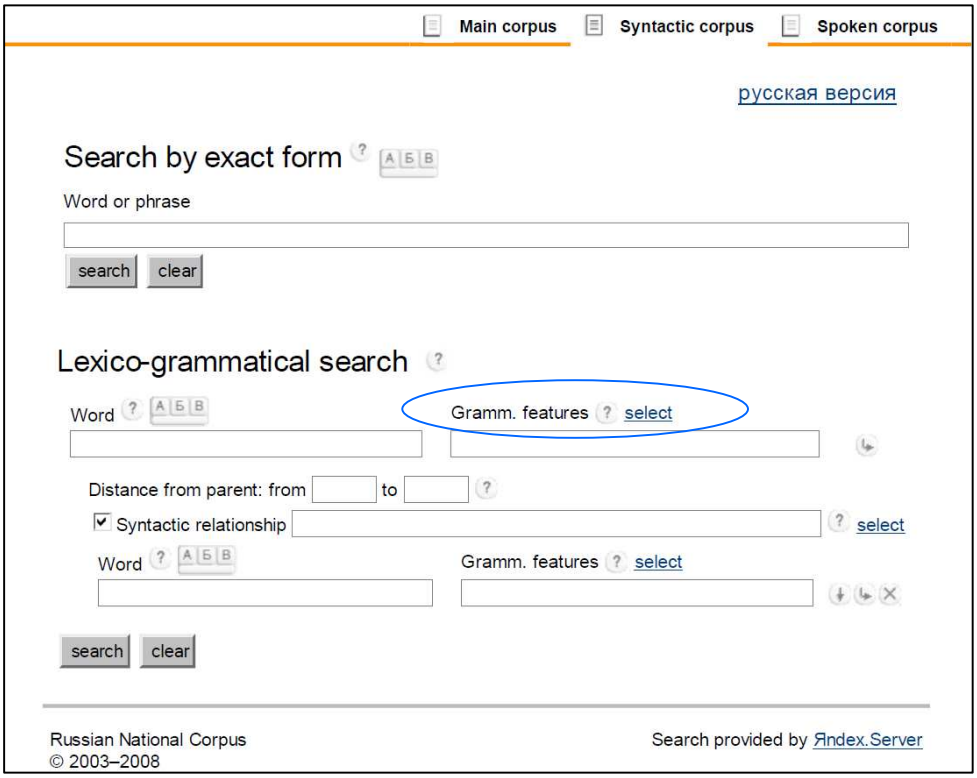
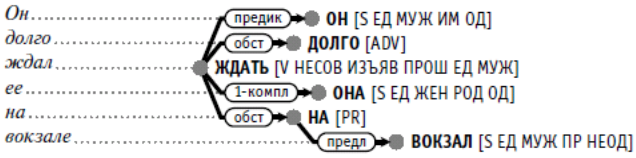


Figure 6: RNC access to the syntactic sub-corpus



The syntactic formalism originates in the Meaning-Text Theory [21], but the inventory of syntactic relations has been extended for the purposes of corpus annotation, incorporating a number of specific linguistic decisions [22, 23].

Figure 7: A sample structure



We, therefore, observe the following straightforward convention when the components of a RNC dependency relation (cf. the inventory in

Figure **8**) are to be mapped on HPSG categories: in a given syntactic dependency relation, the "governor X" corresponds in HPSG to the lexical head of the head daughter, while the "dependent Y" corresponds to the lexical head of the non-head daughter.

As actantial surface syntactic relations connect a predicate word [X] with its syntactic argument [Y], they would by and large map to headed phrases saturating valence requirements. For instance, the first syntactic argument [Y] stands in a predicative, dative-subjective, agentive or quasi-agentive relation to its head [X]. In HPSG, this corresponds to the first position on the head's ARG-ST (argument structure) list, e.g. to the "a-subject". Only in the prototypical predicative relation, however, this also is the single element on the SUBJ (subject) valence list. A non-first syntactic argument [Y] stands in a completive relation to its head [X]. As a rule, the direct object of a transitive verb stands in the first-completive relation to its head while non-transitive single-argument verbs like "sleep", for instance, take no completive relations whatsoever. Eventually, there could be several completive relations, depending on the actual valence requirements of the head. In HPSG this corresponds to the second, third, etc. positions on the head's ARG-ST (argument structure) list. A second large group of surface syntactic relations contains attributive dependencies. These relations connect a word [X] with its dependent word [Y] which functions as a modifier, i.e. is not subcategorized, and by and large would map in an HPSG setup to head–adjunct phrases

As for coordinative constructions, these are conceived in dependency syntax as directed asymmetric relations and in this respect do not stand out from the rest. In an HPSG setup, however, this group of relations would correspond to various types of (non-headed) coordinate phrases. The

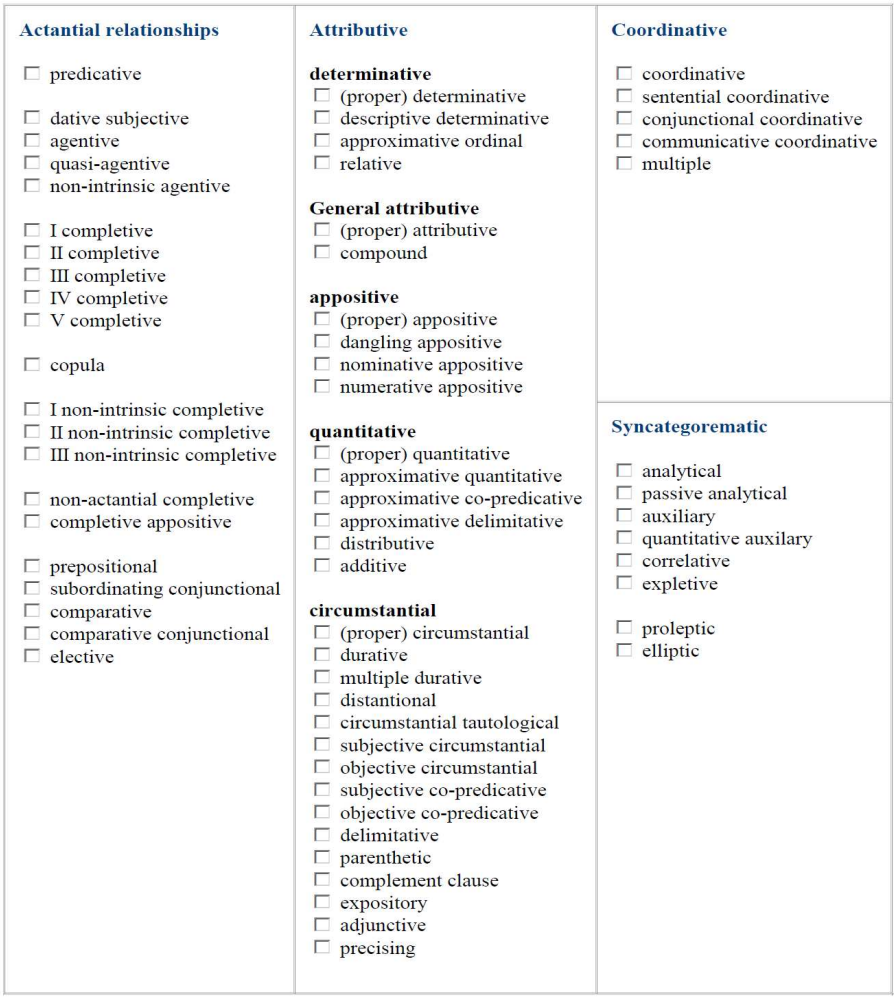
so-called syncategorematic dependencies connect two tightly bound elements [X] and [Y] that are often conceived as intrinsic parts of a larger unit, e.g. of a compound. In an HPSG setup, this group of relations would only partly correspond to headed phrases with functional categories, e.g. auxiliary verbs.

In this valuable resource, even more structured grammatical knowledge is accessible, e.g. with regard to multi-word expressions (MWE), syntactic ellipsis and gapping. The RNC website contains structured lists of orthographically multi-componential lexical units enriched with frequency information from the disambiguated sub-corpus. Based on the collocation analysis and lexicographic resources, two general MWE types are distinguished.

Inasmuch as the components of a MWE can be neither changed nor separated, it is considered equivalent to a single word and represented as a separate node in the syntactic structure. To this first type belong fixed expressions functioning as: (i) prepositions, e.g. по отношению (in relation to); (ii) conjunctions, e.g. коль скоро (as soon); (iii) particles, e.g. разве что (unless), что ни есть (no matter), не то чтобы (not that), нет-нет да и (once in while); (iv) adverbs, e.g. пока что (as yet), как бы то ни было (anyway), чуть ли не (almost), скрепя сердце (reluctantly), из рук вон плохо (thoroughly bad), стало быть (thus), то и дело (time and again), в обнимку (embracing each other), испокон веков (since the beginning of time).

On the other hand, there are syntactically transparent expressions whose components show certain degree of inflectional variation or allow other words to intervene in between. For such a MWE no standard syntactic structure is built, but (some of) its components are combined in an auxiliary dependency relation. It is assumed to hold (from X to Y) in the following examples: сам[Y] себя[X] (oneself); изо[X] дня в[Y] день (from day to day);

Figure 8: Syntactic relations in RNC



так[Y] называемый[X] (so called); все[Y] равно[X] (all the same); знать[Y] не знаю[X] (me having no idea whatsoever); дурак[Y]-то он дурак[X] (him being admittedly a fool).

In elliptical constructions the missing words are reconstructed in the syntactic annotation as “phantom” units which participate in the respective syntactic dependencies without introducing any changes in the original text. Similar approach is adopted in case of gapping, i.e. in constructions with missing verb of “vague” semantic content. An additional empty node is included the dependency structure, with its lemma set to “non-specific verb” assigning it the most plausible characteristics and, based on them, an indication of a lexeme that would represents a “natural hypothesis” for the missing verb.

Having adopted linguistically informed strategies in the modular grammar design, we deliberately concentrate on making the most of the freely available structured grammatical knowledge in the Russian National Corpus. Interfacing with existing corpora and processing tools is therefore fundamental to the Russian Resource Grammar development.

4. Proof-of-concept Implementation

To wrap up, here are some basic figures on the current state of the Russian grammar: ~1000 lines of code (excluding Matrix files and lexicon); ~350 newly introduced types (excluding Matrix types). The invested grammar engineering effort can be estimated as approximately 100 person hours of collaborative grammar development, plus some help from student assistants. Already at this initial stage, the Russian grammar covers the basic word order and agreement phenomena, as well as linking of syntactic to semantic arguments, case assignment by verbs to dependents, ‘pro-drop’ and argument optionality (2, 3, 9), passive (5) and various impersonal constructions (8, 11, 13, 14), among other mosphosyntactic phenomena.

(1) Профессор читает книгу
professor[nom.mask.sg] read[pres.act.3sg] book[acc.fem.sg]
'The professor reads the book.'

(2) Профессор читает
professor[nom.mask.sg] read[pres.act.3sg]
'The professor reads.'

(3) Читает книгу
read[pres.act.3sg] book[acc.fem.sg]
'(pro-drop) reads the book.'

(4) Студент решает задачу
student[nom.mask.sg] solve[pres.act.3sg] task[acc.fem.sg]
'The student solves the task.'

(5) Задача решена
task[nom.fem.sg] solve[pcp.pass.sg.fem]
'The task is solved.'

(6) Профессор дал задачу студентам
professor[nom.mask.sg] give[past.sg.masc] task[acc.fem.sg] student[dat.pl]
'The professor gave the task to the students.'

(7) Вопрос требует особого внимания
question[nom.masc.sg] require[pres.3sg] special[gen.neut.sg] attention[gen.neut.sg]
'The question requires special attention.'

(8) Быстро светает.
quickly dawn[pres.3sg]
'It's dawning quickly.'

(9) Пишем новую статью
write[pres.1pl] new[acc.fem.sg] article[acc.fem.sg]
'We write a new article.'

(10) Президент скоро лишится доверия
presient[nom.masc.sg] soon be-deprived[non-past.3sg] trust[gen.neut.sg]
'The president will soon lose credibility.'

(11) Петра тошнит
Peter[acc.mask.sg] feel-sick[pres.3sg]
'Peter feels sick.'

(12) Старый профессор гордится студентами.
old[nom.mask.sg] professor[nom.mask.sg] be-proud[pres.3sg] student[ins.pl]
'The old professor is proud of the students.'

(13) Быстро темнело.
quickly get-dark[past.sg.neut]
'It was getting dark quickly.'

(14) Отцу нездоровится.
father[dat.masc,sg] feel-unwell[pres.3sg]
'Father does not feel well.

The linguistic analyses encoded in the grammar serve to map the surface strings to semantic representations in Minimal Recursion Semantics (MRS) format [15]. For instance, the MRS in Figure 9 is assigned to the example in (6). It includes the basic propositional structure: a situation of ‘giving’ in which the first argument, or agent, is ‘professor’, the second (recipient) is ‘student’, and the third (patient), is ‘task’. The relations are given English predicate names for the convenience of the grammar developer. A simple tree display in **Figure 10** offers an abbreviated view over the HPSG derivation while hiding the detailed typed feature structures beneath away from the user.

Figure 9: MRS representation.

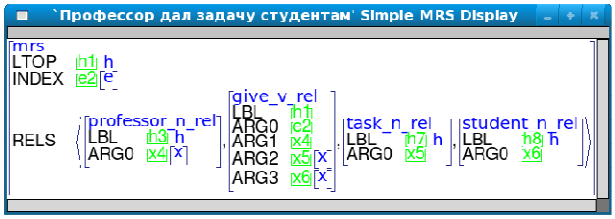
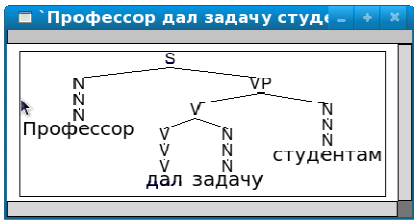


Figure 10: Tree representation



5. Outlook

In the elaboration of the individual grammars and especially for discovering structured linguistic knowledge to be reflected in the respective modules we shall systematically exploit the open access to rich linguistically interpreted corpora available for Slavic languages.

All individual grammars will be designed to support the innovative implementation of a Slavic core module that consolidates strategies for constructing a cross-linguistic resource based on concepts of shared and non-shared morphosyntactic phenomena.

An important desideratum for the individual resource grammars is to eventually couple them with treebanks which either pre-exist or will be constructed in parallel.

6. Acknowledgements

In all these areas, we anticipate international cooperation with distinguished research groups from the Russian Academy of Sciences (Leonid Iomdin), Bulgarian Academy of Sciences (Kiril Simov and Petya Osenova), Polish Academy of Sciences (Adam Przepiórkowski), and others. We envisage for this international exchange to eventually result in an international infrastructural project on Slavic corpora and grammars (SlaviCoGram), and are grateful to all these colleagues for preliminary discussions and constructive cooperation.

References

[1] Uszkoreit, H., D. Flickinger, and S. Oepen, Proposal of Themes and Modalities for International Collaboration on Deep Linguistic Processing with HPSG. 2001, DFKI LT Lab and Saarland University, CSLI Stanford and YY Technologies.

[2] Uszkoreit, H. DELPHIN: Deep Linguistic Processing with HPSG -- an International Collaboration. 2002 http://hans.uszkoreit.net/delphinhome.html.

[3] Uszkoreit, H. New Chances for Deep Linguistic Processing. The 19th International Conference on Computational Linguistics COLING'02. 2002. Taipei, Taiwan.

[4] Bender, E.M., D. Flickinger, and S. Oepen. The Grammar Matrix: An Open-Source-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics. 2002. Taipei, Taiwan.

[5] Pollard, C. and I. Sag, Head-Driven Phrase Structure Grammar. 1994, Chicago: University of Chicago Press.

[6] Copestake, A., Implementing Typed Feature Structure Grammars. CSLI Publications. 2002.

[7] Lønning, J.T. and S. Oepen. Re-usable tools for precision machine translation. COLING|ACL 2006 Interactive Presentation Sessions. 2006. Sydney, Australia.

[8] Oepen, S., et al., LinGO Redwoods. A rich and dynamic treebank for HPSG. Journal of Research on Language and Computation, 2004. 2(4): p. 575-596.

[9] Toutanova, K., et al., Stochastic HPSG parse selection using the Redwoods corpus. Journal of Research on Language and Computation, 2005. 3(1): p. 83-105.

[10] Bender, E.M. and D. Flickinger. Rapid Prototyping of Scalable Grammars: Towards Modularity Extensions to a Language-Independent Core. 2nd International Joint Conference on Natural Language Processing. 2005. Jeju, Korea.

[11] Drellishak, S. and E.M. Bender. A Coordination Module for a Crosslinguistic Grammar Resource. 12th International Conference on Head-Driven Phrase Structure Grammar. 2005. Stanford: CSLI.

[12] Callmeier, U., PET - a platform for experimentation with efficient HPSG processing techniques. Natural Language Engineering, 2000. 6 p. 99-107

[13] Oepen, S., et al. The LinGO Redwoods treebank. Motivation and preliminary applications. The 19th International Conference on Computational Linguistics. 2002. Taipei, Taiwan.

[14] Copestake, A., A. Lascarides, and D. Flickinger. An algebra for semantic construction in constraint-based grammars. The 39th Meeting of the Association for Computational Linguistics. 2001. Toulouse, France.

[15] Copestake, A., et al., Minimal Recursion Semantics: An Introduction. Journal of Research on Language and Computation, 2005. 3(4): p. 281-332.

[16] Flickinger, D. and E.M. Bender. Compositional Semantics in a Multilingual Grammar Resource. ESSLLI Workshop on Ideas and Strategies for Multilingual Grammar Development. 2003.

[17] Flickinger, D., E.M. Bender, and S. Oepen, MRS in the LinGO Grammar Matrix: A Practical User's Guide. 2003.

[18] Bender, E.M. Evaluating a Crosslinguistic Grammar Resource: A Case Study of Wambaya. ACL08 : HLT,. 2008. Columbus, Ohio.

[19] Segalovich, I. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. MLMTA-2003. 2003. Las Vegas.

[20] Sokirko, A., Semantic Dictionaries in Automatic Text Processing (based on materials of the system DIALING) (in Russian). 2007.

[21] Mel'cuk, I.A., The Russian Language in the Meaning-Text Perspective. Wiener slawistischer Almanach. Vol. Sonderband 39. 1995, Moskau - Wien: Gesellschaft zur Förderung slawistischer Studien.

[22] Apresjan, J., et al. A Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects. The fifth international conference on Language Resources and Evaluation, LREC 2006. 2006. Genoa, Italy.

[23] Boguslavsky, I., et al. Development of a dependency treebank for Russian and its possible applications in NLP. The third International Conference on Language Resources and Evaluation (LREC-2002). 2002. Las Palmas