SlaviCLIMB: Combining expertise for Slavic grammar development using a metagrammar

Antske Fokkens[1] and Tania Avgustinova[2]

[1] The Network Institute, VU University Amsterdam,
De Boelaan 1105, 1081 HV Amsterdam, The Netherlands
antske.fokkens@vu.nl,
[2] Department of Computational Linguistics, Saarland University
Campus, 66123 Saarbrcken, Germany
avgustinova@coli.uni-saarland.de

Abstract. We present the initial steps and outlook of SlaviCLIMB, the dynamic component of an effort to develop grammatical resources for Slavic languages around a shared core grammar. The requirements of this component are: (i) it should provide a platform for sharing similar but not identical analyses, (ii) it should be able to contain alternative analyses for the same phenomenon, (iii) it should be easy to use for syntacticians and grammar engineers and (iv) it should provide an interface for Slavicists not trained in grammar engineering allowing them to create grammars. Though the first two properties have been addressed in previous work, this paper is, to our knowledge, the first to address (iii) and (iv) within the context of metagrammars and multilingual grammar development.

Keywords: Metagrammar engineering, alternative analyses management, linguistic hypotheses testing, Slavic languages

1 Introduction

This paper introduces initial ideas on SlaviCLIMB, a metagrammar that is part of an effort to develop HPSG (Pollard and Sag, 1994) grammatical resources for a closed set of languages constituting the traditionally well-studied Slavic language family. The structural design of the applied framework (Avgustinova and Zhang, 2009) is centered on a common Slavic core grammar (SlaviCore) whose components are shared within the language group (Avgustinova, 2007). In the exploration phase of the project, SlaviCore is viewed as an extension of the typologically dedicated components of the LinGO Grammar Matrix (Bender et al., 2002, 2010), with SlaviCLIMB representing a dynamic grammar engineering component that captures language specific variations and facilitates grammar development for individual Slavic languages. We address the main goals of SlaviCLIMB and how we plan to achieve them by creating extensions to existing technology.

We define the following requirements for SlaviCLIMB: It should provide a platform for sharing analyses that are similar, but not identical in different Slavic languages, i.e. it should trigger variations based on parameters of linguistic properties. Second, it should be able to contain alternative analyses for the same phenomenon for more systematic exploration as suggested by Fokkens (2011). Third, it should be easy for syntacticians and grammar engineers to implement analyses. Finally, it should provide a customization interface, so that Slavicists can create grammars for Slavic languages by selecting implemented analyses without being an expert in grammar engineering or even HPSG.

This paper is structured as follows. In §2, we outline the background of this work. §3 illustrates the role of SlaviCLIMB through an example. We describe the currently available software and planned additions for SlaviCLIMB in §4 and conclude by comparing SlaviCLIMB to related work in §5.

2 Background

Limiting the typological coverage to a closed group of related languages provides a special perspective on cross-linguistic modeling. Our long term goal is to encode mutually interoperable analyses of a wide variety of linguistic phenomena, taking into account eminent typological commonalities and systematic differences. The parallel construction of HPSG-based Slavic grammatical resources employs a common core module in combination with language specific extensions and corpus-based grammar elaboration at all stages of the project. Playing a key role in linguistic research, alternative analyses are indispensable in modeling language variation. The Slavic language group offers "laboratory conditions" for experimenting with a novel type of phenomena-oriented alternatives-aware grammar modeling. Therefore, the explicit formulation of alternative analyses needs to be supported from the beginning. As the grammar engineering effort presented here is mainly concerned with enhancing the empirical validation of grammar sharing, the following aspects of multilingual parallel grammar development are of immediate interest to us: (i) what can be included in the common core module and what are the implications of such decisions; (ii) the practicability of phenomena-driven management of alternative analyses; (iii) robust guidelines supporting the grammar engineering activities in (i) and (ii).

The primary purpose of SlaviCLIMB is to support the development of SlaviCore. It follows the CLIMB[3] metagrammar engineering methodology (Fokkens, 2011, Fokkens et al., 2012) which is tightly linked to the Grammar Matrix customization system (Bender et al., 2010). The latter can create starter grammars covering a wide range of typological variation for a limited set of phenomena. Users can specify linguistic properties through a web interface and the customization system automatically generates an implemented grammar. The resulting grammar can then be extended manually. The CLIMB methodology extends the idea of generating code to a general approach for grammar development. The grammar engineer includes new implementations in a metagrammar which uses the original structure of the Grammar Matrix and part of its code generation software. Advantages of the approach include facilitating a phenomenon-based structure of the grammar, supporting alternative analyses for the same phenomenon and increasing modularity (Fokkens et al., 2012). However, we need

[3] Comparative Library Implementations with a Matrix Basis

to introduce a few adjustments to CLIMB in order to accommodate the third and fourth requirement mentioned above, namely: it should be easy to use for grammar engineers, and Slavicists not trained in HPSG should be able to create grammars. These adjustments will be presented in detail in §4. First, we'll illustrate the contribution of SlaviCLIMB through an example in §3.

3 Experimental modeling of the Slavic case system

3.1 The original SlaviCore model

The notion of case, as it is used in linguistic theory, is quite complex and refers to a range of linguistic properties at different levels of description (Avgustinova, 2007, p.25). The main distinction lies in understanding case as morphological marking of forms and as a syntactic notion linked to a particular function. This can be modeled by classifying the type *case* along two dimensions: *f-case* for the level of syntactic functions and *m-case* representing case marking. The functional case is an abstraction over regular case variation and language-specific constraints with respect to case marking. In the HPSG type hierarchy it introduces an additional dimension of case classification. This in particular means that in the lexical entry of a verb, the case value is of type respective f-case type. These abstract case values will be expanded to their concrete instances on the basis of lexical and contextual constraints, taking into consideration the relevant (language-specific) morphological and adpositional case marking.

Currently, SlaviCore contains an elaborate case hierarchy of 117 types that captures the wide variety of case marking found in Slavic languages, as well as (among others) the range of *functional subjective cases* (most subjects in Slavic are nominatives, but Polish also has dative subjects and Russian both datives and genitives, whereas Bulgarian exhibits a morphological base form), as well as *functional objective cases* (mostly accusative, but also genitive and instrumental in Polish and Russian, whereas Bulgarian exhibits a morphological oblique form). None of the Slavic languages uses all 117 types which forms the first motivation for a dynamic component.

3.2 Modeling with SlaviCLIMB

The customization machinery used in CLIMB can add underspecified types to the type hierarchy based on underspecified properties of lexical items or morphological markers.[4] A similar mechanism can be used to create the subtypes that link morphological cases to the right functional cases. As for now, SlaviCLIMB includes an option that triggers the top of the case hierarchy including the main functional cases, while the morphological case can be specified for the language in question. The following addition to SlaviCLIMB will provide a setup where the rest of the type hierarchy is based on direct observations in the language.

[4] This is a Grammar Matrix functionality described in Drellishak (2009).

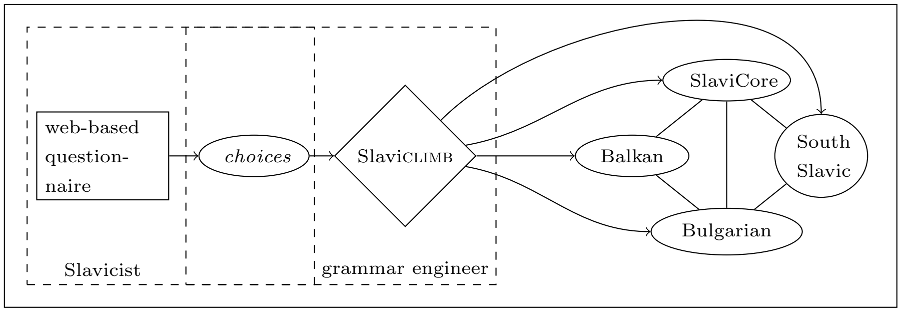

Fig. 1. Overview of development of a Bulgarian grammar with SlaviCLIMB

If a lexical item subcategorizes for a specific functional case, an option indicating the morphological case it subcategorizes for is used to generate the necessary subtypes. In this setup, the case hierarchy depends on the lexical items and morphemes that are defined in *choices*.[5] A Slavicist can define lexical items and morphemes with the relevant linguistic properties and the exact case hierarchy will be derived from these definitions. The case hierarchies thus created for individual languages can be compared to the original proposal in SlaviCore. Modeling the case hierarchy through SlaviCLIMB provides an environment to empirically verify Avgustinova's theoretical work. The original purpose of SlaviCLIMB to support parallel analyses allows grammar writers to either include the functional-morphological distinction or not. For instance, SlaviCLIMB supports Avgustinova's model for Bulgarian placing it in a wider Slavic context, but also the language specific solution adopted in the Bulgarian grammar BURGER (Osenova, 2010).

4 SlaviCLIMB

Figure 1 provides a schematic overview of how a Bulgarian grammar may be created using SlaviCLIMB. The ellipses indicate components that are generated by the system, the rectangle and diamond represent components requiring active interaction with experts. Slavicists fill out a web-based questionnaire, defining phenomena of a particular language and, if applicable, which analysis should be selected. They can indicate whether the phenomenon is a general Slavic property, a property of a subgroup of languages or language specific. This information does not influence the behavior of the ultimate grammar, but is represented in its architecture. The definitions provided by the Slavicist are stored in standardized form in *choices*, which allows us to verify hypotheses about language specific or shared behavior by comparing the *choices* for individual languages. *Choices* is the input to the SlaviCLIMB metagrammar developed by a grammar engineer.

[5] *choices* refers to the file that contains definitions of linguistic properties which trigger implementations in the metagrammar.

The metagrammar creates type definitions based on *choices* and places them in the common core, in a subcore or in the language specific file.

SlaviCLIMB currently covers the process starting with *choices* and can generate the complete SlaviCore and Russian Resource Grammar (RRG, Avgustinova and Zhang, 2010). It supports both the functional-morphological distinction and a simple case hierarchy. As such, it currently fulfills our first two requirements as defined in §1. A declarative version of CLIMB has been developed, which allows grammar engineers to define their analyses in TDL. This revision can serve as a basis to redesign SlaviCLIMB so that it is easier for grammar engineers to work with, which will fulfil the third requirement. It is technically possible for a Slavicist to manually define *choices*, but this requires extensive training. The Grammar Matrix architecture has a web interface that can be filled out by linguists without HPSG expertise, and detailed documentation on how to use the customization system (Fokkens et al., 2012). A Slavic specific version of the interface and documentation will be created to make SlaviCLIMB more accessible to Slavicists (fulfilling the fourth requirement). Designing such an interface is not a trivial task and will be ongoing research requiring both grammar engineering and Slavic expertise. Though the Slavicist will likely need support from grammar engineering experts to fix elements in the grammar or extend it further, a significant amount of work can be done through this method. This is shown by Borisova (2010) who modelled poly-personal agreement in Georgian almost exclusively using the Grammar Matrix customization system.

5 Conclusion

Methods for sharing analyses across grammars have been developed since the early days of grammar engineering. Grammar engineers of individual grammars that are part of the ParGram Project (Butt et al., 2002) meet biannually to compare and discuss their grammars. Kinyon et al. (2006) bring Candito's (1998) MetaGrammar to a cross-linguistic level through a study of verb second word order in several languages. Grammatical Formalism (GF, Ranta, 2009) reveals several similarities to our approach. It uses a metagrammar to divide expertise and share implementations across languages, including family-based cores for Romance and Scandinavian languages. Unlike SlaviCLIMB, it is mainly designed as a practical approach and even though it is based on a well-defined formalism and contains a vast amount of linguistic information, it is not related to a linguistic theory. Both the syntacticians and the engineers with knowledge of the languages in GF need a strong technical background, whereas SlaviCLIMB aims to provide a platform that allows also Slavicists without technical knowledge to create grammars. The CoreGram project (Müller 2013) is particularly interesting in relation to the work presented in this paper. It takes a bottom up approach of sharing analyses between HPSG grammars by means of a common core. If a property is found in several grammars, it is placed in a subcore shared by these grammars. It is in principle possible to gain knowledge about linguistic universals through CoreGram, but no strong claims are made on innateness of such principles. The closed set of languages in the Slavic family allow us to partially adapt a top down approach potentially leading to different insights than a bottom up approach. It should therefore be interesting to compare the insights of the two projects.

The methods mentioned above mainly resemble SlaviCore. To our knowledge, SlaviCLIMB forms the first proposal of a dynamic component that can combine expertise to empirically validate theoretical linguistic models.

References

Avgustinova, T.: Language Family Oriented Perspective in Multilingual Grammar Design. Linguistik International: Band 17. Peter Lang, Frankfurt am Main (2007)

Avgustinova, T., Zhang, Y.: Parallel grammar engineering for Slavic languages. In: Proceedings of GEAF, Singapore (2009)

Avgustinova, T., Zhang, Y.: Conversion of a Russian dependency treebank into HPSG derivations. In: Proceedings of TLT'9. (2010)

Bender, E.M., Flickinger, D., Oepen, S.: The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In: Proceedings of the Workshop on Grammar Engineering and Evaluation, Taipei, Taiwan (2002) 8–14

Bender, E.M., Drellishak, S., Fokkens, A., Poulson, L., Saleem, S.: Grammar customization. Research on Language & Computation **8**(1) (2010) 23–72

Borisova, I.: Implementing Georgian polypersonal agreement through the LinGO Grammar Matrix. Master's thesis, Saarland University (2010)

Butt, M., Dyvik, H., King, T.H., Masuichi, H., Rohrer, C.: The parallel grammar project. In: Proceedings of the workshop on Grammar Engineering and Evaluation. (2002) 1–7

Candito, M.: Building parallel LTAG for French and Italian. In: Proceedings of ACL and ICCL, Montreal, Canada (1998) 211–217

Drellishak, S.: Widespread but Not Universal: Improving the Typological Coverage of the Grammar Matrix. PhD thesis, University of Washington (2009)

Fokkens, A.: Metagrammar engineering: Towards systematic exploration of implemented grammars. In: Proceedings of ACL, Portland, Oregon, USA (2011) 1066–1076

Fokkens, A., Avgustinova, T., Zhang, Y.: CLIMB grammars: three projects using metagrammar engineering. In: Proceedings of LREC 2012, Instanbul, Turkey (2012)

Fokkens, A., Bender, E.M., Gracheva, V.: LinGO grammar matrix customization system documentation (2012)

Kinyon, A., Rambow, O., Scheffler, T., Yoon, S., Joshi, A.K.: The metagrammar goes multilingual: A cross-linguistic look at the V2-phenomenon. In: Proceedings of the 8th International Workshop on TAG and Related Formalisms, Sydney, Australia (2006) 17–24

Müller, S.: The CoreGram project: Theoretical linguistics, theory development and verification. In: Proceedings of the workshop on HMGE. (2013)

Pollard, C., Sag, I.: Head-Driven Phrase Structure Grammar. University of Chicago Press, Chicago, USA (1994)

Osenova, P.: BUlgarian Resource Grammar Efficient and Realistic (BURGER) (2010)

Ranta, A.: The GF resource grammar library. Linguistic Issues in Language Technology **2**(2) (2009)