Reusing a Large Scale Grammar for Multilingual Generation in Slavic Languages

November 17, 1999

Paper ID Code: NAACL-121
Topic Area(s): Multilingual Natural Language Generation
Keywords: natural language generation; grammar development; reuse of existing resources; multilinguality; slavic languages
Submission to NAACL or ANLP (you must choose only one)? NAACL
Word Count: 3148
Under consideration for other conferences (specify)? None

Abstract

The paper discusses the methods followed to re-use a large-scale, broad-coverage English grammar for constructing similar scale grammars for Bulgarian, Czech and Russian for the fast prototyping of a multilingual generation system. We present (1) the theoretical and methodological basis for resource sharing across languages, (2) the use of a corpus-based contrastive register analysis and (3) the possibilities of assessing the resulting grammars. Because the present work concerns the re-use of the grammar of a language that is typologically quite different from the languages treated, the issues addressed in this paper appear relevant to a wider range of researchers in need of large-scale grammars for less-researched languages.

1 Introduction

In this paper we report about our experience in re-using a large-scale, broad-coverage English grammar for building similar scale grammars for Bulgarian, Czech, and Russian. These grammars have been developed for a multilingual text generation system that aims at automatic generation of instructional texts in various styles in the CAD/CAM domain starting from a specification of their meaning.

The motivation for adopting this approach lies in three rather strong constraints:

1. New computational grammars for *three* languages had to be developed. For none of them large-scale computational resources exist.
2. Resource development had to be fast. The time frame for an integrated intermediate prototype was about a year, and grammar development was only one component task.
3. The resources to be built should be easily extensible, i.e., they should not be completely restricted to the sublanguage of the CAD/CAM domain.

Given these constraints, it was necessary to adopt a methodology of resource development which supports *resource sharing* with existing accounts of languages other than the ones that are treated here, and to gear resource development in such a way that a fairly general linguistic account was achieved.

The grammar we have used as a basis is the Nigel grammar of English as implemented in the Penman generator, developed since the 1980s (Mann and Matthiessen1985). The linguistic-theoretical basis of the system is Systemic Functional Linguistics (SFL) (Halliday1985). Many people have contributed to the Nigel grammar, making it into a large-scale grammar of English that covers a very broad range of grammatical phenomena. The most important reason for chosing Nigel as starting point was that it has been claimed to be particularly amenable for re-use and multilingual development. The organisation of the grammar separates specifications of syntactic structures from a description of the functions of those syntactic structures (cf. (Bateman *et al.*1999) and Section 2). (Bateman *et al.*1991) argue that this functional description varies less across languages than does the syntactic description, and, since it is precisely the latter functional component of the description that provides the overall organisation of the grammar, the result is a grammatical account that can serve as a general guideline for the grammatical description of a wide range of languages without enforcing artificial uniformity. The development of generation grammars on this basis is now also supported by an extensive grammar development environment and multilingual sentence generator: Komet-Penman Multilingual system (KPML; (Bateman1997)) which, like Nigel, is available free of charge.

To avoid restriction to the sublanguage of CAD/CAM texts, we compiled a corpus of instructional texts in various styles and genres. The analysis of this corpus confirmed the assumption that a rather wide range of grammatical phenomena had to be covered, if different presentation styles had to be generated.

That generation grammars need to be sufficiently broad in their coverage, if a range of texts is to be produced, is now a widely accepted fact in the NLG community at large. It is also more and more acknowledged that methodologies and tools are needed that allow for a fast development of such grammars, avoiding the necessity of building them from scratch. Methods for re-using existing resources have been proposed and claimed effective in several previous developments and re-use of grammatical resources for closely related languages has been successfully explored in several frameworks (e.g., for the Core Language Engine: (Rayner *et al.*1996), for DATR (Cahill and Gazdar1995)). However, it remains to be shown that employing these frameworks for languages that are typologically quite distinct is still as effective or feasible at all. The availability of a method that can deal with a wide variety of languages, less-researched ones among them, is therefore doubly significant.

This issue is addressed in the present paper. Employing the method of resource sharing as originally proposed in (Bateman *et al.*1991), we provide a further test for this particular development strategy. The Slavic languages dealt with here are typologically quite distinct from English, employing very different modes of organisation. Assessing the extent to which the functional basis of the Nigel grammar remains appropriate in this case is of potential benefit for the development of natural language technology for a large number of still under-represented languages.

2 Method of resource development

In the present section we describe the theoretical principles of cross-linguistic resource sharing (Section 2.1) and the contrastive-linguistic corpus analysis (Section 2.2) carried out prior to implementation (Section 2.3). The approach advocated here is a combination of a *system-oriented* method of grammar development and an *instance-oriented* one. 'System-oriented' means building up a computational resource with a view to the whole language system; 'instance-oriented' means being guided by a register analysis.

2.1 Resource sharing

The idea of *resource sharing* across languages was originally applied to English, Chinese, Japanese, German and Dutch (Bateman *et al.*1991; Teich *et al.*1996) and is implemented in KPML. The theoretical basis of the concept of resource sharing in KPML is Systemic Functional Linguistics (SFL) (Halliday1985; Halliday and Matthiessen1999). SFL enjoys a number of properties that make it particularly suitable for modeling multilingual resources. Comparing any two languages, one will always detect commonalities and differences. The general representational constructs employed in SFL can be used as parameters along which such commonalities and differences can be described in a principled way. SFL is a functional theory of language, in which the concept of function is reflected in three *metafunctions*. The ideational metafunction is concerned with states-of-affairs and their circumstances (processes and participants involved, such as Actor, Goal, and adjuncts of Time, Space etc). The interpersonal metafunction is to do with the role relations of speaker and hearer in a discourse (grammatical mood and modality). The textual metafunction represents the patterns with which cohesive and coherent texts are created. Other organising principles are *stratification, axiality, delicacy* and *ranking*. The strata distinguished are lexico-grammar, semantics, and context. Linguistic description, at each stratum, has two aspects, one representing linguistic *systems* (paradigmatic axis), the other the *structural realisations* of these systems (syntagmatic axis). The means used for the representation of the paradigmatic axis is the system network. Formally, a system network is a type hierarchy supporting multiple inheritance. The types (or: features) have attributes expressing constraints on syntagmatic realisation. For instance, English mood consists of the types 'declarative', 'interrogative' and 'imperative'. The type 'declarative', for example, has the attributes '+Subject' and 'Order-Subject-before-Finite'. Thus, the functional, systemic aspects are separate from the formal, syntagmatic ones. Furthermore, a system network adheres to a partial ordering which is called *delicacy*, denoting the type-subtype relation. Taking again the example of an English mood description, the subtypes of 'interrogative', 'polar-interrogative' and 'wh-interrogative', are more delicate than 'interrogative'. System networks are set up for different *ranks* (clause, groups/phrases, words, morphemes), where the sets of features holding for each rank are pairwise disjoint. Stratification, metafunction, axiality, delicacy and rank make up the basic inventory of representational categories used in SFL. Applying these categories to cross-linguistic description, the following observations can be made.

1. Languages tend to show more similarities on the more abstract **strata** of linguistic organisation than on the less abtract ones (i.e., they tend to express similar meanings, but cast them in different grammatical terms). For example, all languages express the semantic category of *speech function*, i.e., make statements, ask questions, give orders. However, the grammatical potential realising these semantic categories may be quite different across languages.

2. Languages tend to be similar on the paradigmatic **axis** and less similar in terms of syntagmatic realisation. A case in point is the expression of definiteness in the nominal group in Bulgarian and English (examples under (1) in Figure 1). In English, definiteness is expressed by the definite determiner and in Bulgarian by suffixation of the first element in the nominal group.

3. Systems of low **delicacy** tend to be similar across languages, and systems of higher delicacy tend to be dissimilar. An example is mood in English and Russian (examples under (2) in Figure 1). The basic mood options of English also apply to Russian. However, regarding imperatives, Russian further distinguishes according to formalness: there is one verb form for formal imperatives (2.a) and one for informal imperatives (2.b). That is, Russian has more delicate options for imperative than English. The same holds for Bulgarian and Czech.

4. There may be different preferences in different languages concerning the grammatical **rank** at which a particular meaning is expressed. A case in point is the signalling of referents in a discourse as identifiable or nonidentifiable. While in Bulgarian and English identifiability is marked in the nominal group by

(1) Definiteness

En: *Now specify the multiline.*

Bu: *Sega zadaite multiliniata.*
Now specify-FORMAL-IMP multiline-DEFINITE

(2) Imperative mood

En: *Click on the OK button.*

Ru:

a. *Nazhmite knopku OK.*
Click-FORMAL-IMP button-ACCUSATIVE OK

b. *Nazhmi knopku OK.*
Click-INFORMAL-IMP button-ACCUSATIVE OK

(3) Headings

En: *To draw a polyline*

Bu: *Chertane na polilinija*
drawing-NOMINALISATION of polyline

Cz: *Nakreslení křivky*
drawing-NOMINALISATION polyline-GENITIVE

Ru: *Chtoby narisovatj poliliniju*
in-order-to draw-INFINITIVE polyline-ACCUSATIVE

Figure 1: Examples

determiners, in Czech and Russian it is typically marked by word order in the clause.

5. Different languages may distribute functional responsibilities differently across **metafunctions**. The same formal means may serve different functions. For example, the passive construction is possible in Bulgarian, Czech, Russian and English. In English the main function of the passive is to change the Theme-Rheme organisation, so passive carries a textual function. Czech and Russian, on the other hand, can simply change Theme-Rheme by changing word order and the passive thus carries less of a textual load.

KPML implements these five types of cross-linguistic variation and goes thus well beyond comparable approaches in multilingual computational linguistics. First, it is possible to represent *both* commonalities *and* differences between languages. In most computational approaches to multilinguality, one or the other is taken as central. For instance, most machine translation approaches would either enforce commonality at one particular level (interlingua-based approaches), or assume divergence as the basic principle (transfer-based approaches). Second, with the SFL model as a basis, we achieve a *multi-dimensional* model of cross-linguistic variation, in which that variation can be defined along the parameters presented above. The types of cross-linguistic divergences commonly considered are much less refined and do not come as close to covering the observable cross-linguistic phenomena, such as the ones presented above.[1]

2.2 Contrastive-linguistic register analysis

The second pillar in the approach to a fast prototyping of grammatical resources for generation pursued here is working from *instances*, i.e., from a corpus of instructional texts. Its goal was to identify the repertoire of the most common lexico-grammatical constructions in the three languages in this register.

Considering the results of previous work on lexical and syntactical closure of sublanguages (e.g., (Kitteredge1987)), a small corpus appeared sufficient. The corpus (consisting of a set of parallel texts from the CAD/CAM domain and a set of non-parallel instructional texts from other domains) was analysed following (Hartley and Paris1996), tagging for the type of text unit (e.g., procedure, description), plan element (e.g., goal, side-effect, step), and a set of SFL lexico-grammatical features.

The analysis revealed the following tendencies of similarities and differences between the linguistic resources needed for each language:

- Top-level goals are typically expressed by nominalisations in Bulgarian and Czech and by non-finite clauses in Russian.
- The actions to be performed are realised mostly by imperatives in Russian and Bulgarian. In Czech indicative is also used (25%).
- The actions are mainly material-directed processes, in Bulgarian and Czech also mental and relational processes occur.
- The similarity of the three languages is very high with respect to the features positive, active and non-modal (more than 90%). Differences arise mainly in finiteness: Bulgarian and Czech tend to employ

[1]Cf. e.g., the kinds of translational divergences proposed by (Dorr1994).

mostly finite forms, while Russian employs 32% infinite forms. In all three languages, the finite forms are predominantly in second person plural, which corresponds to the formal imperative form.

- The Czech corpus shows a strong preference for simple clauses (65%). In the Bulgarian and Russian corpora the majority of clauses are complex, mainly hypotactic, while in Czech paratactic and hypotactic relations are evenly distributed.

The corpus analysis identified the repertoire of the most common constructions in this register in each language and revealed a large overlap across the three languages. These two factors were made use of in the resource implementation.

2.3 Implementation

The implementation of grammars for Bulgarian, Czech and Russian proceeded in two stages.

Stage 1 built upon the grammar of English. The register analysis identified the grammatical systems to be dealt with to cover the grammatical phenomena of the register. Resource development was distributed according to language.

Implementation by resource sharing proceeds in cycles of "copying" and "adapting" grammatical systems. For instance, the system accounting for predicate types in systemic functional grammars is TRANSITIVITY. For English, this system distinguishes between material, mental, relational and verbal predicates, each with a number of subtypes and particular syntagmatic realisational constraints. In the first step of implementation, the grammar developer has to assess whether the basic systemic options apply to the new language. In a lot of cases, this will be true. The system can then be taken over as it is. Typically, what has to be changed are the attributes of the system's types, i.e., the constraints on structural realisation. For instance, the attributes of English material processes are simply that there are two participants Actor and Goal. For Russian and Czech we have to add the constraints that Actor will be realised in nominative case and Goal in accusative case (for active voice); for Bulgarian, nothing has to be added because there is no morphological case. The larger part of implementation comes down to these two actions, 'copy' and 'adapt', which are supported by the KPML environment, and the new resource can be written out as the grammar of the new language, inheriting from English. The fact that the larger part of grammar implementation for a new language consists of these two actions supports the hypothesis that languages are more similar on the paradigmatic *axis* than in terms of the realisation of paradigmatic options.

These actions are repeated going depth-first, i.e., going further in *delicacy*, until the point of coverage that is required by the sublanguage. Here, according to the hypothesis that languages tend to be similar in terms of less delicate system and different in terms of more delicate ones, it becomes less likely that systems can still be shared. For instance, for Bulgarian, Czech and Russian more delicate systems had to be added for imperatives (cf. examples under (2)). Those parts of a grammatical system that are not needed for the sublanguage are left in place, so that the grammar remains an executable specification.

Stage 2 of grammar implementation was organised according to linguistic phenomena rather than languages. Given that at least Czech and Russian are typologically closer to each other than to English, there are common linguistic constructions that are better to treat copying from Russian to Czech or vice versa than copying from English. The grammars now inherit less from English and more from each other. Phenomena that were not covered adequately in the English grammar because they are not so prominent in the language are treated focally in this stage. Cases in point are syntactic agreement, which is rather complex in all of Bulgarian, Czech and Russian, aspect (a verbal category specific to Slavic languages), and word order, which is determined primarily pragmatically in Czech and Russian. Also, in this stage of development, the language-specific requirements of the sublanguage are treated. For instance, Russian section headings are realised by purpose clauses, whereas in Bulgarian and Czech they are realised by nominalisations (examples under (3) in Figure 1).

With the combination of resource sharing as implemented in KPML and a multlingual corpus-based register analysis, a maximally contrastive-linguistic approach to grammar implementation for Bulgarian, Czech and Russian has been made possible.

3 Valuating the scale of the achievement

Benchmarking the grammatical coverage of an NLG system is a different proposition from benchmarking the coverage of a Natural Language Understanding (NLU) system. Even if the concern is solely with the task of surface realisation, and then with only the output from that task and not its inputs (Dale and Mellish1998), the enterprise remains challenging.

Yet from the perspectives of both linguistic theory and applications potential, we need to find some measure of our achievements which will permit comparisons with other available grammars and with corpora on which they might be exercised. NLU has progressed towards this goal by developing test suites, as in the TSNLP project (Lehman et al. 1996), (Oepen et al. 1997). The availability of a restricted suite of items appears advantageous for NLG also. This applies particularly to the lexicon, where the connections to the underlying semantics may be complex. However, no such resources exist for the Slavic languages studied here. Moreover, we need to ask in how far it is sensible to adopt test suites developed as input to NLU tasks as "target suites" for measuring features of the output of an NLG task.

While test suites contain negative data and constructions that are hard - such as elliptical constructions - or potentially ambiguous, target suites should exclude such items, since they would constitute inappropriate output for a generator. On the other hand, test suites may under-represent phenomena that are important or difficult for NLG. For example, TSNLP accords very little weight to extragrammatical phenomena, such as parentheticals and temporal expressions. Test items may also need to be designed to systematically represent markers of rhetorical relations or anaphoric expressions (Yeh & Mellish1997). The items in a target suite would ideally have semantic annotations which allowed the extraction of sets of items which are closely related semantically - "functionally" in terms of SFL, or "paraphrastically" in terms of MTT (e.g. (Nasr et al. 1998)). Undergeneration is not necessarily such a problem for NLG as under-recognition is for NLU.

So the relation between coverage and application potential is more complex than it first seems. Using current test suites to benchmark the coverage of NLG systems would give very misleading results. However, the advantages of appropriate target suites are clear for developers and users alike. The challenge is now to design them. As a first step, we should consider what a benchmark for NLG should look like - what is it that an NLG test suite should *test*? Secondly, a test suite could be started by annotating the output of the system developed here, using a TSNLP-compatible metalanguage together with tools developed by the DiET project (Netter et al. 1998).

4 Conclusions

The approach employed here combines two methods of computational grammar implementation for the purpose of a fast prototyping of computational linguistic resources. The approach builds upon one particular interpretation of re-usable resource, which is the idea of *resource sharing across languages.* We called this a *system-oriented* method of creating computational resources for new languages (Section 2.1). Complementary to this method, we have been guided by a corpus-based contrastive register analysis of Bulgarian, Czech and Russian instructional texts, i.e., an *instance-oriented* method of computational resource development (Section 2.2). Proceeding this way, basic *general-language grammars* and *sublanguage grammars* for CAD/CAM instructional texts have been created. Due to the system-orientation, these grammars are less restricted than sublanguage grammars of a particular domain; and due to the instance-orientation, these grammars are adequate for the domain at hand as well. Also, this two-tiered resource development method allowed us to practice a contrastive-linguistic approach and a maximal sharing of efforts, not only working from English, but also sharing among the three Slavic languages (Section 2.3).

The method exercised here is of a more general potential interest in contexts where resources for more than one language have to be implemented in a short period of time and where no resources for typologically similar languages exist that could be used as a basis. In future work, the achievements of this strategy in the present scenario will be assessed along the lines suggested in Section 3.

References

John A. Bateman, Christian M.I.M. Matthiessen, Keizo Nanri, and Licheng Zeng. The re-use of linguistic resources across languages in multilingual generation components. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence, Sydney, Australia*, volume 2, pages 966 – 971. Morgan Kaufmann Publishers, 1991.

John A. Bateman. Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering*, 3(1):15–55, 1997.

John A. Bateman, Christian M.I.M Matthiessen, and Licheng Zeng. Multilingual natural language generation for multilingual software: a functional linguistic approach. *Applied Artificial Intelligence*, 13:607–639, 1999.

Lynne J. Cahill and Gerald Gazdar. Multilingual lexicons for related languages. In *Proceedings of the 2nd DTI Language Engineering Conference*, pages 169–176, London, 1995. Department of Trade and Industry.

Dale, R. and Mellish, C. Towards evaluation in natural language generation. In *Proceedings of the first international conference on Language Resources and Evaluation*, Granada 1998 pp. 555-562

Bonnie J. Dorr. Machine translation divergences: a formal description and proposed solution. *Computational Linguistics*, 20(4):597–634, December 1994.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. *Construing experience through meaning: a language-based approach to cognition.* Cassell Academic, London and New Work, 1999.

Michael A.K. Halliday. *An Introduction to Functional Grammar.* Edward Arnold, London, 1985.

Anthony Hartley, Danail Dochev, Nevena Gromova, Kamenka Staykova, Alla Bémová, Aleksandr Rosen, Jiří Trojánek, and Elena Sokolova. Tagging and analysis of instructional texts in the software domain. Technical report, ITRI, University of Brighton, 1998. AGILE deliverable CORP (CORP-Bu, CORP-Cz, CORP-Ru).

Hartley, A. and Paris C. Two Sources of Control over the Generation of Software Instructions. Information Technology Research Institute Technical Report Series, ITRI-96-02. 1996

Kasper, Robert. SPL: A Sentence Plan Language for text generation. Unpublished manuscript. University of Southern California (USC)/Information Sciences Institute (ISI), 1989

Kitteredge R. The Significance of Sublanguage for Automatic Translation. In: Nirenburg S. (ed.) *Machine Translation: Theoretical and Methodological Issues.* 1987

Lehmann, S. et al. TSNLP – Test Suites for Natural Language Processing. In *Proceedings COLING*, 1996 pp. 711-716

William Mann and Christian M.I.M. Matthiessen. A demonstration of the Nigel text generation computer program. In James D. Benson and William S. Greaves, editors, *Systemic Perspectives on Discourse, vol 1.* Ablex, Norwood, N.J., 1985.

Nasr, A., Rambow, O. and Kittredge, R. A linguistic framework for controlled language systems. *Procedings of the second international Controlled Language Applications Workshop CLAW'98*, Pittsburgh 1998

Netter, K. et al. DiET - Diagnostic and Evaluation Tools for natural language processing applications. *Proceedings of the first international conference on Language Resources and Evaluation*, Granada 1998 pp. 573-579

Oepen, S., Netter, K. and Klein, J. TSNLP – Test Suites for Natural Language Processing. In: J. Nerbonne (ed.) *Linguistic databases*, CSLI Lecture Notes, Stanford: CSLI, 1997

M. Rayner, D. Carter, and P. Bouillon. Adapting the Core Language Engine to French and Spanish. In *Proceedings of NLP-IA-96*, Moncton, new Brunswick, May 1996.

Sparck-Jones, K. and Galliers, J.R. *Evaluating natural language processing systems.* Berlin: Springer-Verlag, 1996

Elke Teich, Liesbeth Degand, and John A. Bateman. Multilingual textuality: Experiences from multilingual text generation. In G. Adorni and M. Zock, editors, *Trends in Natural Language Generation: an artificial intelligence perspective*, number 1036 in Lecture Notes in Artificial Intelligence, pages 331–349. Springer-Verlag, Berlin, New York, 1996.

Yeh, C-L. and Mellish, C. An empirical study on the generation of anaphora in Chinese. *Computational Linguistics* 23:1, pp. 169-190, 1997