Adapting SProUT to processing Baltic and Slavonic languages

Witold Drożdżyński* **Petr Homola**† **Jakub Piskorski*** **Vytautas Zinkevičius**‡
German Research Center for Artificial Intelligence
Saarbrücken
Germany
{drozdzynski,phomola,piskorsk}@dfki.de, vytasz@ktl.mii.lt

Abstract

This paper focuses on presenting an initial effort for porting SProUT — a novel general purpose IE platform, to processing Baltic and Slavonic languages. We describe the system, characterize the mentioned language groups and discuss the process of developing named-entity and chunk grammars for these languages, which are crucial for solving information extraction tasks.

1 Introduction

Prompt, sound and timely information is an essential factor in competition in business of any kind. Recent advances in information technology such as Information Extraction (IE) provide dramatic improvements in conversion of the vast amount of raw textual information in digital form in a myriad of data repositories on Intranets and Internet into valuable and structured data. The task of IE is to identify instances of a particular pre-specified class of entities, events and relationships in natural language texts, and the extraction of the relevant arguments of the identified events or relationships (Appelt & Israel 99; Pazienza 99). Such information is prerequisite for discovering more complex patterns in textual data collections and is widely used for boosting other information management technologies, such as Q/A systems, search engines, text mining, and text summarization.

There has been a vast bulk of research in the area of building efficient and high coverage IE systems for English and a few other major languages (Hobbs *et al.* 97; Humphreys *et al.* 98; Aone & Ramos-Santacruz 00; Ciravegna *et al.* 00). However, relatively few efforts have been undertaken for building and adapting IE platforms for processing Slavonic and Baltic languages which are highly inflectional and exhibit relatively free word-order. (Maynard & Cunningham 03) and (Humphreys *et al.* 02) present some work on adapting the IE modules of the famous GATE platform (Cunningham *et al.* 02) for processing Bulgarian and Russian. A comprehensive description of adapting unification-based formalism for processing Polish is described in (Przepiórkowski *et al.* 02). With respect to Czech, some relevant work has been presented in (Holub & Míka 01).

This paper focuses on presenting an initial effort for adapting SProUT[1] (Becker *et al.* 02) — a novel general purpose IE platform, to processing Czech, Polish, and Lithuanian. In particular, we report on the crucial issues concerning extending and adapting SProUT's existing resources to these languages, and we take an insight into named-entity and chunk grammar development.

The rest of this paper is organized as follows. Section 2 gives an overview of the Baltic and Slavonic language families. The main facts concerning SProUT and the underlying grammar formalism are introduced in section 3. In section 4, the available NE-grammars are described. The issues concerning integration of morphological analyzers for targeted languages and adapting the existing grammars to these languages are addressed in section 5 and 6, respectively. Further, grammar extensions are described in section 7. Section 8 gives some examples and finally, in section 9 we draw some conclusions and discuss future work.

2 Baltic and Slavonic languages

Slavonic languages are a large group of the Indoeuropean language family. The languages split further to West, East and South Slavonic subgroups. All Slavonic languages are quite similar, not only lexically, but also typologically.[2] They

* German Research Center for Artificial Intelligence (DFKI), Saarbrücken

† Institute of Formal and Applied Linguistics (ÚFAL), Charles university, Prague

‡ Institute of Mathematics and Informatics, Vilnius

[1] Shallow Text Processing with Unification and Typed Feature Structures

[2] Except of Bulgarian and Macedonian, which are somewhat different.

have rich inflection and a free word order.[3] The inner structure of noun phrases, which are the object of our analysis, are similar to the German noun phrases (cf. (Eroms 00)), except that Slavonic languages do not have the article (except of Bulgarian and Macedonian).

Baltic languages form a small group of the Indoeuropean language family. Nowadays, only Latvian and Lithuanian are being spoken.[4] Baltic languages are typologically similar to Slavonic languages, although the vocabulary is rather different. They have rich inflection, free word order, up to ten cases (the Lithuanian dialect spoken in Belarus, cf. (Ambrazas 96)). The verbal system is very complicated, e.g., Lithuanian has eight tenses (cf. (Ambrazas 96)). The verbal inflection allows to express many modalities (debitive, optative, modus relativus etc., cf. (Forssman 01)). Baltic adjectives and participles (and some pronouns and numerals) can have a special definite form, which is similar to the use of the article in German (cf. (Forssman 01)), especially in Latvian; the slightly different usage of these forms in Lithuanian can help to determine named entities and collocations in merely morphologically annotated texts.[5]

With respect to SProUT, are current activities centered around Czech and Polish (both West Slavonic) and Lithuanian (East Baltic).

3 SProUT

In this section we briefly describe SProUT (Becker *et al.* 02), a platform for the development of multilingual IE components. An Achilles heel of most of the earlier IE systems was the fact that they were either lacking efficiency or expressiveness of the underlying grammar formalism. The main motivation for developing SProUT centers around finding a trade-off between these two crucial features. The grammar formalism used in SProUT is a mixture of finite-state techniques which are known to be very efficient and unification-based formalisms which on the other hand guarantee transparency and expressivity. A grammar consists of pattern/action rules, where the LHS of a rule is a regular expression over typed feature structures (TFS) with functional operators and coreferences, representing the recognition pattern, and the RHS of a rule is a seqeuence of TFS, specifying the output structure. Coreferences provide a stronger expressiveness since they create dynamic value assignments and serve as means of information transport into the output descriptions. Other IE frameworks (among others GATE) provide so called dynamic variables instead of coreferences, which can be accessed on the RHS in so called annotation manipulation statements for output production, which call native code. Hence, rule writing in such formalism is done in somewhat less 'declarative' manner and is more difficult for non-programmers. The example grammar rule in SProUT for recognition of location PPs illustrates the syntax.

```
loc-pp :>
 morph & [POS Prep & #preposition,
  INFL [CASE #1, NUMBER #2, GENDER #3 ]]
 morph & [POS Det,
  INFL [CASE #1, NUMBER #2, GENDER #3 ]] ?
 (morph & [POS  Adj,
  INFL [CASE #1, NUMBER #2, GENDER #3 ]] ) *
  gazetteer & [ TYPE general_location ,  SURFACE #location]
  -> phrase & [CAT location-pp,PREP #preposition, LOCATION #location].
```

The first TFS matches an item with part-of-speech Preposition. Then one or zero Determiner items are matched. Subsequently zero or more adjectives are consumed. Finally, the last TFS matches an item in a location gazetteer. The variables #1, #2, and #3 establish coreferences between features which constrains agreement in case, number, and gender for all but last matched items. The RHS enforces creation of a TFS of type phrase, where the matched preposition and location are transported into the corresponding slots via the variables #preposition and #location. For optimizing such extended finite-state networks with rich label descriptions, a bag of methods has been developed which go beyond standard finite-state techniques (Krieger & Piskorski 03).

Currently, the system is equipped with a set of prefabricated and reusable online processing components for basic operations such as tokenization, morphological analysis, and gazetteer lookup, where corresponding linguistic resources for the major Germanic, Romance and Asian languages are provided. Since TFSs are used as a uniform I/O data structure they can be coupled

[3] The word order is not restricted grammatically, except of the inner structure of noun phrases and the position of the clitics, but it is not arbitrary, of course — the criteria of the word order are dominated by the topic/focus articulation (cf. (Sgall *et al.* 80)).

[4] Extinct Baltic languages are: (Old) Prussian, Jatvingian, Curonian.

[5] Slavonic languages have two adjectival forms as well, but their function is different, they do not express the contextual boundness (in the sense of (Sgall *et al.* 80)) of noun phrases, but their syntactic function.

in order to form a system instance straightforwardly by defining a regular expression over their names (Krieger 03). In this manner, highly cascaded architectures can be instantiated.

4 Multilingual NE-grammar

Some work has already been accomplished towards developing named-entity grammars in SProUT for the major Germanic and Romance languages, with the emphasis on maximal re-use of linguistic resources across different languages (Bering *et al.* 03). In order to tackle this challenging task, token classes, some gazetteers, output structures, and grammar fragments are shared for different languages. In particular, SProUT's ability for defining cascaded rules which can be distributed among numerous files, facilitated the idea of shared grammars.

Since the core element structures in the aforementioned languages are identical, generic grammar files were introduced. Let us consider as an example the fact that unknown organization names usually consists of one or more words, which start with an initial capital letter, or consist solely of upper case letters, or include at least one upper case letter (see the rule general_unknown_org in the grammar fragment below). On the other hand , structures like *Siemens AG*, *Siemens S.A.*, or *University of Sheffield* are covered by language-specific grammars since designators following or preceding organization names are language specific. The following simplified grammar fragment illustrates this example. The rule en_org_suffix specifies company designators for English (analogously en_org_university), whereas the rule en_unknown_organization extends the general rule (general_unknown_org) with language-specific context.[6]

```
general_unknown_org :>

 (token & [TYPE first_capital_word, SURFACE %org]
 | token & [TYPE all_caps_words, SURFACE %org]
 | token & [TYPE mixed_word_first_capital, SURFACE %org])+
 -> dummy.

en_org_suffix :>

 (token & [SURFACE "LTD"]
  token & [SURFACE ".", ID "Ltd." & #en_org_type]
  | token & [SURFACE "PLC", ID "PLC" & #en_org_type]
  | token & [SURFACE "Corporation", ID "Corporation"
  & #eng_org_type]) -> dummy.

en_org_university :>

 morph & [STEM "university", ID "univeristy" & en_org_type]
 morph & [SURFACE "of"] ? -> dummy.

en_unknown_org :>

 ( @seek(general_unknown_org)
   @seek(en_org_suffix)
   | @seek(en_org_university)
     @seek(general_unknown_org) )

 -> enamex & [ORGNAME %org,
              ORGTYPE #en_org_type,
              DESIGNATOR nil]
```

In this way, the process of writing and editing grammars appears to be less laborious since potential adjustments or extensions might simply require modifications of the language-independent generic grammar units. The provided language-specific grammars can be adopted to a new language by changing the keywords (e.g., designators) and by providing static named-entity lists for the gazetteer.

5 Morphological components for Balto-Slavonic languages

Although SProUT contains a module for creating morphological lexicons based on full form word lists, we integrated external components for Baltic and Slavonic languages. The main reason is that these languages are highly inflectional, thus a lexicon containing all possible word forms would contain millions of records. Moreover, morphological disambiguation can be used if desired.

The Czech analyzer and tagger has been developed at ÚFAL (Hajič 01). The program uses a positional system of 15 tags (e.g., AAFS1—-1A—- means adjective, feminine gender, singular, nominative, positive degree and not negated), but less tags are used usually for a particular word form (for example, substantives and adjectives do not have tense, verbs do not have degree etc.). The participles are marked as adjectives (they have all categories of adjectives, except of the degree in some cases, which may be excluded semantically). The lexicon of the analyzer contains more than 800 000 lemmas, which means that it is able to recognize several millions of word forms. The meaning of some tag values depends on other tags occasionally. For example, the tag for gender can have as its value **Q**, which means *feminine* for singular, but *neuter* for plural, i.e., the meaning depends on the tag for number, thus the post-processing is not that straightforward. Furthermore, a statistical tagger for disambiguation is available.

[6] ”%org” - the symbol % represents a special (weaker) type of coreference under Kleene star, where all values are collected in a list which is then transported to the RHS of the rule.

Morfeusz, the Polish analyzer, has been developed at the Institute of Computer Science of the Polish Academy of Sciences. It returns a sequence of tags (Przepiórkowski & Woliński 03) for each word form (or a set of tag sequences, if the word form is ambiguous), which has to be postprocessed, similarly as the Czech results.

For Lithuanian, we integrated *Lemuoklis* (Zinkevičius 01), a morphological analyzer, lemmatizer and tagger in one. It assigns to each token its lemma (or several hypothetical lemmas) as well as all potential morphological analysis. A word form is characterized grammatically by a combination of properties with respect to 13 categories: part of speech, aspect, reflexiveness, voice, mood, tense, group, degree, definiteness, gender, number, case and person. The database of lexical and grammatical information of the program consists of six lexicons (organized as letter trees). Three of them store roots of Lithuanian words, which are associated with appropriate morphological rules. Two others store word forms with no morphological information. The last one contains a list of abbreviations and acronyms. In "Lemuoklis", morphological rules of the inflectional word changing are expressed in form of digital tables. The tables represent graph structures that define both collections of affixes and grammatical properties. Using morphological rules together with word-root lexicons enables to analyze grammatically milliards of theoretically available Lithuanian written forms. In case a surface form is homonymous, i.e., it has several grammatical meanings, the program gives a full grammatical characteristic for each possible homoform of the surface form. However, some methods are used to reduce the ambiguity without taking into account the context. One of them is disambiguation between diminutive nouns that have flexion -yti(s) and respective verbal infinitive forms. The disambiguation between proper and common nouns is performed by utilization of special lexicons containing proper forms from Lithuanian corpora and other sources. Forms with shortened endings are quite common in Lithuanian texts. Analogously, these forms are recognized by means of special lexica that were primarily designed for spell-checking in Lithuanian.

The morphological analyzers are implemented as servers, thus once a user has configured the SProUT environment (host/port name), he can use them independently of the operating system that he uses.

6 Adaptation of available resources

Let us turn to adapting NE-grammars for Czech, Polish and Lithuanian. It is quite obvious that this adaptation requires many changes in the rules, because the typology of the target languages is different. The most important difference is, of course, rich inflection. The available German grammars served as the source for producing grammars for targeted languages. The generic grammars have been fully taken over, as expected, but adapting language specific rules introduced some difficulties.

The declension of substantives in contemporary German is not very rich, many forms in the paradigm are identical due to syncretism. The same holds, even more, for proper names. Therefore, the attribute SURFACE (i.e., the word form, as it occurred in the input text) is used mostly in the rules (directly or over the gazetteer word list). This approach is, in general, not applicable to Balto-Slavonic languages, since the set of possible word forms of a lemma is usually much larger. Hence, the major changes centered around using the attribute STEM (i.e., the basic word form) instead of the attribute SURFACE and providing additional information to control the inflection. For example, the rules, which recognize simple person names (given name followed by a family name), look in the German grammar as follows:

```
given_name:> Gazetteer & [TYPE given_name, SURFACE #given_name]
  -> dummy.

family_name:> Gazetteer & [TYPE family_name, SURFACE #surname]
  -> dummy.

person_name1:> @seek(given_name) @seek(family_name)
  -> ENAMEX & [TYPE "PERSON",
  surname #surname,
  given_name #given_name].
```

Obviously, adapting these rules directly would require the presence of all possible inflected forms for each name, e.g., in Czech *Petr, Petra, Petrovi, Petře* etc. Since such solution is not practicable, the attribute STEM is used and additional conditions are defined to ensure proper agreement. This is illustrated in the following example:

```
person_name1 :>
  Gazetteer & [TYPE given_name, STEM #given_name,
    CASE_NOUN #case, NUMBER_NOUN #number]
  Gazetteer & [TYPE family_name, STEM #surname,
    CASE_NOUN #case, NUMBER_NOUN #number]
  -> ENAMEX & [TYPE "PERSON" ,
    surname #surname,
    given_name #given_name].
```

Other phenomena, e.g., time expressions, are processed similarly. We changed analogously almost all rules (about 90%), the attribute SURFACE remained only for uninflected words (such as particles, prepositions etc.). Currently, the grammars consist of 96 rules for each language.

Additionally, the use of the attribute STEM in combination with the gazetteer is related with an essential problem: in SProUT, words recognized by the gazetteer are not processed morphologically. This fact does not impact processing German or English, due to their poor inflection, but this is obviously not sufficient for processing languages with rich inflection. A simple solution would be to generate the full declension paradigm for every word in the word list. Thus, there would be several lists, for each relevant combination of morphological tags. Although this ad hoc solution would not affect the computational complexity significantly (typically only the endings are different, the size of the corresponding finite-state representation would only slightly grow), it is not very elegant. An analogous solution could be realized by using SProUT's functional operators, which serve as predicates or for constructing additional information. We suggest another solution, namely to process the input text morphologically and then to look up in the gazetteer for values of the attribute STEM. This improvement ensures that the word would be recognized and provided with the necessary morphological tags.

Let us briefly mention another issue, which affects the process of adapting the grammars: the differences in morphological annotation. This problem is rather technical than linguistic, but it is essential for the grammarians. We use different morphological analyzers (described in section 5), which use their own morphological tags that are not compatible with SProUT's generic tagsets. This is caused mainly by various linguistic viewpoints (or linguistic tradition of the languages) and partially by decisions of the author of the morphological software. For example, participles are annotated by the Czech analyzer as adjectives, but the Lithuanian analyzer marks them as verbs. In other words, the Czech analysis outputs an independent lemma for participles, whereas the Lithuanian analysis gives the lemma of the verb the participle is derived from. We decided to convert the output of the morphological analysis without significant changes of tags and to leave this technical issue to the grammar developers, who are expected to be familiar with the design of the morphological components, although it means slightly more work during the adaption. Moreover, as there is no universal solution, we rely upon the decision of the designers of the morphological components. The Czech grammars served as the source for creating analogous grammars for Polish and Lithuanian (the typology is very similar, minor differences concern cases and word order).

7 Grammar extensions

We extended also the NE-grammars described in the previous section to partial syntactic analysis of noun phrases.

Noun phrases in Czech, Polish and Lithuanian have similar inner structure, which is determined by quite strict rules (unlike the order of head participants and adjuncts, which depends mainly on the information structure in these languages). We developed a simple grammar that analyzes simple noun phrases. The following list gives the types of syntactic constructions recognized by our grammar (with examples):

Adjectival attributes *opóźniony pociąg pośpieszny* “late fast train” (Pol), *operační systém* “operating system” (Cze), *baltosios naktys* “white nights” (Lit);

Numeric attributes *dvacet pět let* “twenty five years” (Cze);

Pronominal attributes *tento systém* “this system” (Cze), *aname puslapyje* “on that web page” (Lit);

Genitive attributes *universiteto rektorius* “rector of the university” (Lit), *ředitel banky* “director of the bank” (Cze);

Prepositional attributes *kniha na stole* “book at the table” (Cze), *aikštė miesto centre* “square in the centre of the town” (Lit);

Appositions *prezidentas Paksas* “president Paksas” (Lit);

Adverbs *za duża rzeka* “a too big river” (Pol), *per lėtas traukinys* “a too slow train” (Lit), *světle zelené auto* “light green car” (Cze);

Participles *rychle napsaný dopis* “fastly written letter” (Cze), *iš tolo matomas gaisras* “from far observable fire” (Lit);[7]

The syntactic structure of noun phrases can be generally much more complex. For example, we ignore embedded sentences and participles with actants, thus not all syntactic dependencies are recognized. We use a similar approach as described in (Žáčková 02). However, even this analysis can be wrong, for example, if a noun phrase in genitive follows another noun phrase, which is not the head of it, a dependency (which does not exist in this case) will be proposed. Unfortunately, it is not possible to solve this kind of errors in the shallow approach, a deeper analysis would be needed.

8 Sample rules and analysis

In this section, we present some sample rules in the SProUT notation and a sample analysis.

The following rule is one of the rules used for recognizing date expressions:

```
date_phrase :>
  @seek(dofm)
  ( @seek(month_word)
  | @seek(month_number))
    @seek(year)
  -> TIMEX & [ TYPE "POINT",
      DOFM #dofm_id ,
      MONTH #month_id ,
      YEAR #year_id ].
```

The embedded rules are defined as follows:

```
dofm:> Gazetteer & [TYPE daynr, SURFACE #dofm_id]
  token & [SURFACE "."]
  ->  dummy.

month_word:> Gazetteer & [TYPE month-cz-long,
  STEM #month_id, CASE_NOUN #case]
  -> dummy.

month_number:> Gazetteer & [TYPE month-cz-short,
  SURFACE #month_id]
  token & [SURFACE "."]
  -> dummy.

year:> Gazetteer & [TYPE yearnr, SURFACE  #year_id]
  | ( token & [SURFACE "’"] ?
    Gazetteer & [TYPE yearnr-short, SURFACE #year_id]  )
  ->dummy.
```

The rule *date_phrase* can be utilized, for example, in the following rules:

```
date_point1 :> @seek(date_phrase)
  -> TIMEX & [ TYPE "POINT",
      DOFM #dofm_id ,
      MONTH #month_id ,
      YEAR #year_id,
      CASE #case & "genitive" ].

date_point2 :> token & [SURFACE "k"] @seek(date_phrase)
  -> TIMEX & [ TYPE "POINT",
      DOFM #dofm_id ,
      MONTH #month_id ,
      YEAR #year_id,
      CASE #case & "dative" ].
```

The first rule recognizes expressions like *1. května 2003* “at May 1” (Cze), the second rule recognizes expression with the preposition *k*, e.g., *k 1. květnu 2003*, which has the same meaning. The result is the following feature structure in both cases:

$$(1) \quad \begin{bmatrix} \text{TYPE} & \text{”POINT”} \\ \text{DOFM} & \text{”1”} \\ \text{MONTH} & \text{”květen”} \\ \text{YEAR} & \text{”2003”} \\ \text{CASE} & \textit{case} \end{bmatrix}$$

The structures differ only in the value of the attribute CASE, which is not relevant from the semantic point of view.

The following examples concerns the grammar for noun phrases. An example of a simple Czech noun phrase is (2):

(2) *operačního systému* “operating system”

The output of the morphological components looks as follows:[8]

$$(3) \quad \begin{matrix} \begin{bmatrix} \text{SURFACE} & \text{‘operačního’} \\ \text{STEM} & \text{‘operační’} \\ \text{POS} & \textit{adjective} \\ \text{GENDER} & \textit{masc} \\ \text{CASE} & \textit{genitive} \\ \text{NUMBER} & \textit{sg} \\ \text{DEGREE} & \textit{positive} \end{bmatrix} \\ \begin{bmatrix} \text{SURFACE} & \text{‘systeḿu’} \\ \text{STEM} & \text{‘systém’} \\ \text{POS} & \textit{noun} \\ \text{GENDER} & \textit{masc} \\ \text{CASE} & \textit{genitive} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \end{matrix}$$

Because the necessary agreement in relevant morphological categories between both words holds, the rule for adjectival attributes creates a new structure (4) (we leave out the morphological attributes):

$$(4) \quad \begin{bmatrix} \text{SURFACE} & \text{‘systeḿu’} \\ \text{STEM} & \text{‘systém’} \\ \text{POS} & \textit{noun} \\ \text{ATTR} & \left\langle \begin{bmatrix} \text{SURFACE} & \text{‘operačního’} \\ \text{STEM} & \text{‘operační’} \\ \text{POS} & \textit{adjective} \end{bmatrix} \right\rangle \end{bmatrix}$$

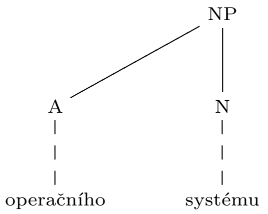
The structure (4) represents the syntactic structure of the noun phrase given in (5) (the vertical edge marks the head of the phrase, the

[7] Only participles without actants are recognized. There is no separate rule for participles in the Czech grammar, because the morphological component marks participles (except of the perfect) as adjectives.

[8] For the sake of simplicity, we give only essential attributes in the feature structures.

oblique edge the dependent; constituent structures in this notation can be converted to a dependency structure by contracting all vertical edges to one node).

(5)



The next example (in Polish) shows, how genitive attributes are analyzed:

(6) *Instytut podstaw informatyki*
“Institute of the foundations
of computer science”

The output of the morphological component is:

$$(7)\quad \begin{bmatrix} \text{SURFACE} & \text{‘instytut’} \\ \text{STEM} & \text{‘instytut’} \\ \text{POS} & \textit{noun} \\ \text{GENDER} & \textit{masc} \\ \text{CASE} & \textit{nom} \\ \text{NUMBER} & \textit{sg} \end{bmatrix} \begin{bmatrix} \text{SURFACE} & \text{‘podstaw’} \\ \text{STEM} & \text{‘podstawa’} \\ \text{POS} & \textit{noun} \\ \text{GENDER} & \textit{fem} \\ \text{CASE} & \textit{genitive} \\ \text{NUMBER} & \textit{pl} \end{bmatrix} \begin{bmatrix} \text{SURFACE} & \text{‘informatyki’} \\ \text{STEM} & \text{‘informatyka’} \\ \text{POS} & \textit{noun} \\ \text{GENDER} & \textit{fem} \\ \text{CASE} & \textit{genitive} \\ \text{NUMBER} & \textit{sg} \end{bmatrix}$$

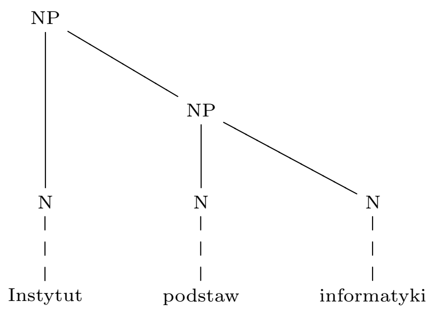
The rule for genitive attributes checks the POS tags and the case of the dependents, and constructs the following structure:

$$(8)\quad \begin{bmatrix} \text{SURFACE} & \text{‘instytut’} \\ \text{STEM} & \text{‘instytut’} \\ \text{POS} & \textit{noun} \\ \text{GEN-ATTR} & \left\langle \begin{bmatrix} \text{SURFACE} & \text{‘podstaw’} \\ \text{STEM} & \text{‘podstawa’} \\ \text{POS} & \textit{noun} \\ \text{GEN-ATTR} & (*) \end{bmatrix} \right\rangle \end{bmatrix}$$

$$(*) = \left\langle \begin{bmatrix} \text{SURFACE} & \text{‘informatyki’} \\ \text{STEM} & \text{‘informatyka’} \\ \text{POS} & \textit{noun} \end{bmatrix} \right\rangle$$

The structure (8) represents the constituent structure (9):

(9)



9 Conclusions and future work

We have reported on an initial effort of adapting the SProUT system and its currently available resources for processing Baltic and Slavic languages. In particular, Czech, Polish and Lithuanian were investigated. External morphological components have been integrated due to the highly inflectional character of these languages. The existing German grammars have been utilized for creating analogous grammars for the languages we focused on. Extensions concerned mainly ensuring the correct agreement between constituents (such as in date expressions). Additionally, noun phrase grammars have been developed.

The result of the syntactic analysis is only partial. It is not possible to get a perfect analysis, because a lot of ambiguities would require a deeper approach (see (Žáčková 02)), which goes beyond the current capabilities of SProUT. A possible solution is to use a stochastical tagger before applying the grammars (we are using a tagger for Czech), but wrong tag assignments would be propagated to the grammar processing level.

Our future work will include extending the implemented grammars and adapting the system to other Slavonic languages. Due to the typological proximity of Balto-Slavonic languages and our initial experiments, we expect that the system could be extended easily to other members of this language family. Particularly, Russian, Slovene and Croatian are being considered. Finally, we intend to integrate the morphosyntactic tagger for strongly inflective languages presented in (Dębowski 03).

Acknowledgements

We are greatly indebted to Marcin Woliński who provided morphological analyzer for Polish and did some implementation work in the integration process. Further, we would like to thank Jan Hajič for the Czech tagger. The research reported in this paper was supported by the German Federal Ministry for Education, Science, Research, and Technology under the grant no. 01 IN A01 (COLLATE).

References

(Ambrazas 96) V. Ambrazas. Dabartinės lietuvių kalbos gramatika. *Mokslo ir enciklopedijų leidykla, Vilnius*, 1996.

(Aone & Ramos-Santacruz 00) C. Aone and M. Ramos-Santacruz. RESS: A large-scale relation and event extraction system. *In Proceedings of ANLP 2000, Seattle, USA*, 2000.

(Appelt & Israel 99) D. Appelt and D. Israel. An introduction to information extraction technology. *A Tutorial prepared for IJCAI Conference*, 1999.

(Becker *et al.* 02) M. Becker, W. Drozdzynski, H.U. Krieger, J. Piskorski, U. Schaefer, and F. Xu. SProUT - Shallow Processing with Typed Feature Structures and Unification. *In Proceedings of ICON 2002, Mumbai, India*, 2002.

(Bering *et al.* 03) C. Bering, W. Drożdżyński, G. Erbach, C. Guasch, P. Homola, S. Lehmann, H. Li, H.-U. Krieger, J. Piskorski, U. Schaefer, A. Shimada, M. Siegel, F. Xu, and D. Ziegler-Eisele. Corpora and evaluation tools for multilingual named entity grammar development. *Proceedings of the International Workshop: Multilingual Corpora - Lingusitic Requirements and Technical Perspectives, Lancaster, UK*, 2003.

(Boehmová & Holub 00) A. Boehmová and M. Holub. Use of dependency tree structures for the microcontext extraction. *In: Proceedings of the Workshop on Recent Advances in Natural Language Processing and Information Retrieval (ACL 2000), pp. 23–33*, 2000.

(Ciravegna *et al.* 00) F. Ciravegna, A. Lavelli, and G. Satta. Bringing information extraction out of the labs: The Pinocchio environment. *In Proceedings of ECAI 2000, Berlin, Germany*, 2000.

(Cunningham *et al.* 02) H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. *In Proceedings of ACL 2002, Philadelphia, USA*, 2002.

(Dębowski 03) Ł. Dębowski. A reconfigurable stochastic tagger for languages with complex tag structure. *Proceedings of EACL 2003, Budapest, Hungary*, 2003.

(Eroms 00) H.W. Eroms. Syntax der deutschen Sprache. *Walter de Gruyter, Berlin*, 2000.

(Forssman 01) B. Forssman. Lettische Grammatik. *Verlag J.H. Roell, Dettelbach*, 2001.

(Hajič 01) J. Hajič. Disambiguation of rich inflection (computational morphology of Czech). *Karolinum, Charles University Press, Prague*, 2001.

(Hobbs *et al.* 97) J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. FASTUS - A cascaded finite-state transducer for extracting information from natural language text. *In Finite-State Language Processing, E. Roche and Y. Schabes, MIT Press, Cambridge, MA*, 1997.

(Holub & Míka 01) M. Holub and P. Míka. MATES — an experimental linguistic database system. *Proceeding of the IRCS Workshop on Linguistic Databases, University of Pennsylvania, Philadelphia*, 2001.

(Homola 02) P. Homola. Machine translation among Slavic languages. *In: Proceedings of the WDS, Charles University, Prague*, 2002.

(Humphreys *et al.* 98) K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. University of Sheffield: Description of the LaSIE-II system as used for MUC-7. *In Proceedings of MUC-7*, 1998.

(Humphreys *et al.* 02) K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. Slavonic named entities in GATE. *Research Memorandum CS-02-01, University of Sheffield, Great Britain*, 2002.

(Krieger & Piskorski 03) H.U. Krieger and J. Piskorski. Speed-up methods for complex annotated finite state grammars. *DFKI Report*, 2003.

(Krieger 03) H. U. Krieger. SDL - A description language for building NLP systems. *Proceedings of the HLT-NAACL 2003 Workshop - Software Engineering and Architecture of Language Technology Systems, Edmonton, Canada*, 2003.

(Maynard & Cunningham 03) D. Maynard and H. Cunningham. Multilingual adaptations of a reusable information extraction tool. *In Proceedings of EACL 2003, Budapest,Hungary*, 2003.

(Pazienza 99) M. Pazienza. Information extraction: Towards scalable, adaptable systems. *Lecture Notes in Computer Science 1714 Springer 1999, ISBN 3-540-66625-7*, 1999.

(Pecina & Holub 02) P. Pecina and M. Holub. Sémanticky signifikantní kolokace. *Technical report TR-2002-13, ÚFAL/CKL, Faculty of Mathematics and Physics, Charles University, Praha*, 2002.

(Przepiórkowski & Woliński 03) A. Przepiórkowski and M. Woliński. A flexemic tagset for Polish. *Proceedings of Morphological Processing of Slavic Languages, EACL 2003, Budapest*, 2003.

(Przepiórkowski *et al.* 02) A. Przepiórkowski, A. Kupść, M. Marciniak, and A. Mykowiecka. Formalny opis języka polskiego. *Book, Akademicka Oficyna Wydawnicza, ISBN 83-87674-35-4, Warsaw, Poland*, 2002.

(Sgall *et al.* 80) P. Sgall, E. Hajičová, and E. Buráňová. Aktuální členění věty v češtině. *Studie a práce lingvistické, Academia, Praha*, 1980.

(Žáčková 02) E. Žáčková. Parciální syntaktitcká analýza (češtiny). *PhD thesis. Fakulta informatiky Masarykovy univerzity, Brno*, 2002.

(Zeman 01) D. Zeman. How much will a RE-based preprocessor help a statistical parser? *Proceedings of the Seventh International Workshop on Parsing Technologies, Tsinghua University Press, ISBN 7-302-04925-4, Beijing Daxue, Beijing*, 2001.

(Zinkevičius 01) V. Zinkevičius. Lemuoklis — morfologinei analizei. *Darbai ir dienos*, 2001.