A Prague Markup Language Profile for the SemTi-Kamols Grammar Model

Lauma Pretkalniņa, Gunta Nešpore, Kristīne Levāne-Petrova, Baiba Saulīte
Institute of Mathematics and Computer Science
University of Latvia
Raiņa bulv. 29, Rīga, LV-1459, Latvia
{lauma,gunta,kristine,baiba}@ailab.lv

Abstract

In this paper we demonstrate a hybrid treebank encoding format, derived from the dependency-based format used in Prague Dependency Treebank (PDT). We have specified a Prague Markup Language (PML) profile for the SemTi-Kamols hybrid grammar model that has been developed for languages with relatively free word order (e.g. Latvian). This has allowed us to exploit the tree editor TrEd that has been used in PDT development. As a proof of concept, a small Latvian treebank has been created by annotating 100 sentences from "Sophie's World".

1 Introduction

Two general approaches can be distinguished in the syntactic representation: the phrase structure approach (Chomsky, 1957) and the dependency approach (Tesnière, 1959). Dependency grammars are usually treated and implemented in a simplified way, if compared to Tesnière's original approach, sacrificing the linguistic details for the benefit of efficient parsing algorithms (Jarvinen and Tapanainen, 1998). In the result, each running-word is treated as a separate part of sentence, which is involved in a separate dependency relation. The SemTi-Kamols hybrid dependency grammar for Latvian implements and extends Tesnière's basic concepts (Bārzdiņš et al., 2007; Nešpore et al., 2010).

Manual development of a Latvian treebank (according to the SemTi-Kamols model) would be very laborious and the tool support is crucial. The SemTi-Kamols model is based on the dependency approach, therefore we have chosen to adapt the annotation tool TrEd (Hajič et al., 2001) that has been proven itself developing the Prague Dependency Treebank (Hajič et al., 2000). The SemTi-Kamols model is more complex than that of PDT analytical layer, as we use both dependencies and phrase structures in the same tree.

TrEd itself is a rather generic-purpose tree editor that can be customized to specific treebank requirements by providing an appropriate extension module. The main component of such a module is a schema that describes the data format. The module also contains style sheets specifying how the data should be represented visually. It may contain some macros for additional support as well — to automate the common annotation tasks or to detect common annotation errors.

2 SemTi-Kamols model

Apart from dependency links, the SemTi-Kamols model is based on a concept of "x-word": a syntactic unit describing analytical word forms and relations other than subordination (Bārzdiņš et al., 2007; Nešpore et al., 2010). From the phrase structure perspective, x-words can be viewed as non-terminal symbols, and as such substitute (during the parsing process) all entities forming respective constituents. From the dependency perspective, x-words are treated as regular words, i.e., an x-word can act as a head for depending words and/or as a dependent of another head word. The following constructions are treated as x-words:

- analytic forms of a verb, e.g. the perfect tense;
- numerals (e.g. *trīsdesmit trīs* 'thirty three') and other multi-word units;

Bolette Sandford Pedersen, Gunta Nešpore and Inguna Skadiņa (Eds.)
NODALIDA 2011 Conference Proceedings, pp. 303–306

- prepositional phrases;
- coordination etc.

3 Data format

Our data format is specified in the XML-based Prague Markup Language (PML). PML is the default input format for TrEd; it is also the main data format of PDT (Pajas and Štěpánek, 2006).

We have adapted the multi-layer annotation approach from PDT (Hajič et al., 2000; Pajas and Štěpánek, 2006). PDT has four annotation layers: *w*, *m*, *a*, *t*. At the *w or* word level, text is divided in tokens and paragraphs. The *m* or morphological level adds morphological annotations and spelling error corrections. At the *a* or analytical level syntactic annotations (dependency links) are added. The top level is the tectogrammatical level *t*, which contains semantic annotations. All the levels (their nodes) are connected through unique IDs. In this paper we address only the first three levels.

The first level (*w*) is taken from PDT as is. The second level (*m*) is adapted with minor changes. We use the possibility to annotate spelling mistakes in the source text at this level. We use most of PDT spellchecking categories and we have added one more to indicate that two tokens form one morphological unit.

The third level (*a*) is the most interesting case. In PDT all relations between parts of sentence are represented using dependencies only, while for the SemTi-Kamols model we need more sophisticated means to deal with both dependencies and phrase structure components (x-words). Further we will examine our *a*-level tree structure.

To operate with a PML document in TrEd, it is necessary to specify which elements correspond to the nodes of the parse tree to be drawn on TrEd's pane, as the rest of the elements describe attributes of these nodes. The tree structure itself corresponds to the (tree) structure of the PML/XML document. The possible structure of the document also needs to be described. It is done by providing the corresponding PML schema to TrEd. PML elements are linked with tree nodes by adding the attribute "role" (with values "root", "node", "childlist" etc.) to the definitions of appropriate elements in PML schema.

Here the first issues arise, as TrEd supports nodes with only one child-list. However, we would like to create a scheme, where each node can have two types of children. One type would represent dependants, the other type — constituents of parent node (this is the case of an x-word). Each node would be able to have any number of children of any of those types. Also, there must be a simple way, how human-annotator can change whether the particular node is parent's dependant or constituent from TrEd. To achieve this, all the children must have the same node type definition in the PML schema. It seems that the only reasonable solution to handle nodes with both types of children is to use artificial nodes.

For each node we introduce one optional child of a special kind — a "container for constituents". If parent node has no container node for constituents as a child, all the children are parent's dependants (see fig. 1). If there is such a container node, its siblings are considered as parent's dependants, but the container node's children — as constituents of the container node's parent. If the node has the container node as a child, there is no token from text, corresponding to this node; in this case, no tokens correspond to the container nodes, too. On the one hand, this makes our PML schema more complicated, but, on the other hand, this significantly improves its usability for a human-annotator.

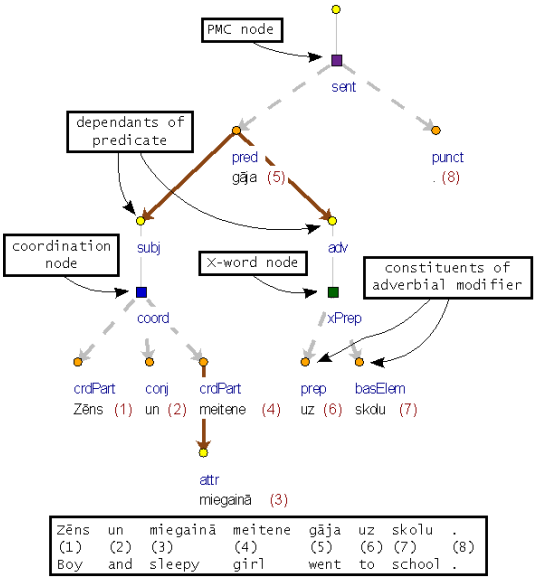

Figure 1: Tree for sentence *Zēns un miegainā meitene gāja uz skolu* 'The boy and the sleepy girl went to the school'

The distinction has been made between three types of containers for constituents. One type is coordination (both coordinated parts of sentence and clauses), other type is so-called genuine x-words (x-words mentioned above other than coordination); the last type is PMC nodes. PMC

(punctuation mark constructs) are the phrase-like systems which hold together some subtrees with corresponding punctuation marks.

PMC is a novelty in attribution to SemTi-Kamols model. As in Latvian the punctuation represents the grammatical structure of the sentence, showing it in the syntax tree is significant to create comprehensive model for the sentence. Nonetheless, to interpret PMC as fully eligible phrase structure would be inadequate in relation syntax theory of Latvian, as PMC components have far more flexible structure as x-words or coordination. PMC nodes handle punctuation marks for constructions like direct speech, subordinate clauses, insertions and parenthesis etc.

Distinction between coordination and genuine x-words was made to make SemTi-Kamols model closer to the original Tesnière's model.

For the dependent children we denote their syntactic roles. For the constituent children we denote their function in the phrase they constitute. We hope, this will facilitate detection of inconsistent markup avoiding issues mentioned by Boyd et al. (2008). For each container node for constituents we add a tag showing the type of x-word (e.g., x-predicate or x-preposition), coordination or PMC. For x-words and coordinate parts of sentence we provide a tag similar to those used at the morphological level. This tag describes the function carried out in the sentence by the whole unit.

Every token in a sentence (even punctuation marks) corresponds to some node in the tree, but not all the nodes have corresponding tokens. As mentioned above, the container nodes for constituents and their direct parents have no corresponding tokens, but there is one more case with no corresponding token. We handle omitted parts of sentence using nodes with no corresponding tokens, for example, elliptical predicate is displayed as "empty" node with additional tag. In all other ways these nodes act as normal nodes — they can have both dependants and constituents.

4 Additional support

We have developed an extension module for TrEd to enable TrEd to work with the trees described above. This extension contains not just schemas, but also helper macros and style sheets.[1]

[1] Module is provided under GPL and can be downloaded here http://eksperimenti.ailab.lv/tred/

We developed two basic ways for visual representation of the trees from Latvian Treebank. One way is the Full view (Fig. 2). It is created to be used for annotators, and it displays every single node as it is, and adds red warnings to the nodes that have probably incorrect roles.

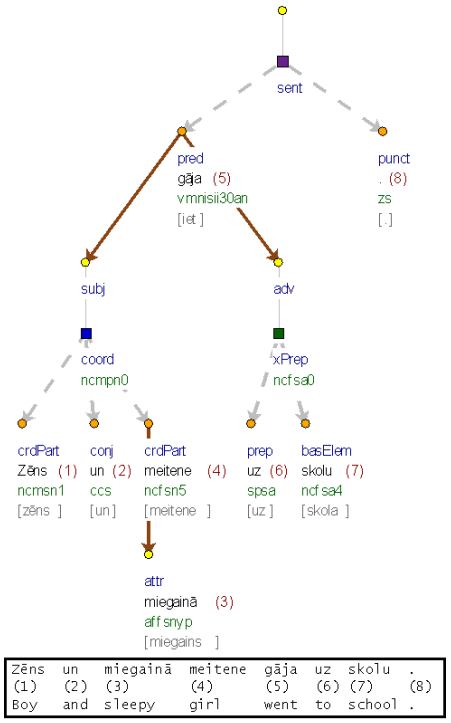

Figure 2: A sentence in the Full view with grammatical information

The other way is the Compact view. It is created to be used for end-users of corpora who don't want to be buried in technical complexities yet need to have full access to all the data. In the Compact view (Fig. 3) container nodes for constituents are displayed as differently colored edges from their parent to their children, thus obtaining the representation we wanted in the beginning of interaction with TrEd. Also there is a possibility to choose, whether to show the grammatical information — lemma and tag. The Compact view can't be used to edit trees.

TrEd implicitly validates data against given PML schema. TrEd does not permit editing, which leads to incompatibility with schema. These features act as a simple error preventing mechanism. As PML schema is not all-powerful we have developed additional macros to check easy-detectable deviations from the intended tree structure. In most cases detected deviations are mistakes made by annotator, but in some cases this was the way to discover incompleteness in our intended structure.

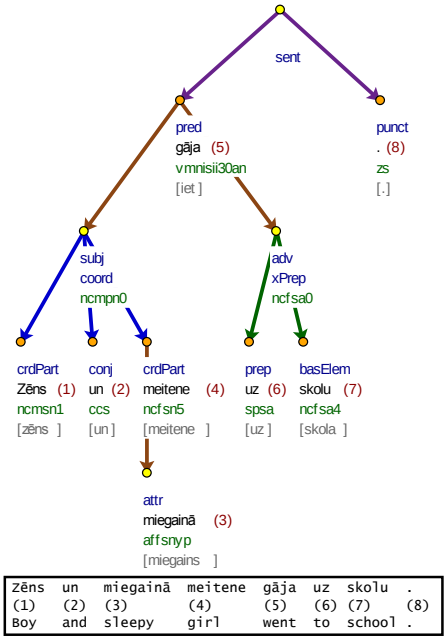

Figure 3: A sentence in the Compact view with grammatical information

5 "Sophie's World"

As a proof of concept, we have annotated first 100 sentences of J. Gaarder's "Sophie's World" using the developed infrastructure.

Annotation was done as follows. First, the morphological markup was added in a semi-automated way. After that, linguist trained in work with TrEd manually created preliminary trees. Finally, trees where discussed and verified by general meeting consisting from 2 or 3 linguists and the architect of PML schema for Latvian Treebank. This multi-step process allowed us to repeatedly verify whether the intended schema and data format is appropriate for the Latvian language, whether it can represent all the encountered phenomena of the language, whether the later added schema additions is consistent with the initial intentions.

6 Conclusion

The integration of PDT tools and SemTi-Kamols' grammar model so far has proved to be successful and should be continued by integrating PDT tools with the rule-based SemTi-Kamols' partial parser (Bārzdiņš et al. 2007). The next step would be to develop a bigger treebank to cover all the syntactic constructs of Latvian and to obtain more precise results and statistical information to build a statistical parser. Though, even the 100 sentences annotated so far covers most of syntactic constructions typical for standard Latvian.

Acknowledgments

This work is funded by the State Research Programme "National Identity" (project No 3) and the Latvian Council of Sciences project "Application of Factored Methods in English-Latvian Statistical Machine Translation System".

Reference

Guntis Bārzdiņš, Normunds Grūzītis, Gunta Nešpore, Baiba Saulīte. 2007. *Dependency-Based Hybrid Model of Syntactic Analysis for the Languages with a Rather Free Word Order*. Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA), pp. 13–20

Adriane Boyd, Markus Dickinson, Detmar Meurers. 2008. *On Detecting Errors in Dependency Treebanks*. Research on Language and Computation 6(2), pp. 113–137.

Noam Chomsky. 1957. *Syntactic Structures.* The Hague: Mouton

Jan Hajič, Alena Böhmová, Eva Hajičová, Barbora Vidová Hladká. 2000. *The Prague Dependency Treebank: A Three-Level Annotation Scenario.* A. Abeillé (ed.): Treebanks: Building and Using Parsed Corpora, Amsterdam: Kluwer, pp. 103–127.

Jan Hajič, Barbora Vidová Hladká, Petr Pajas. 2001. *The Prague Dependency Treebank: Annotation Structure and Support.* Proceedings of the IRCS Workshop on Linguistic Databases, Proceedings of the IRCS Workshop on Linguistic Databases, Philadelphia, USA, pp. 105–114.

Timo Järvinen, Pasi Tapanainen. 1998. *Towards an implementable dependency grammar.* Proceedings of the Workshop on Processing of Dependency-Based Grammars, pp. 1–10.

Gunta Nešpore, Baiba Saulīte, Guntis Bārzdiņš, Normunds Grūzītis. 2010. *Comparison of the SemTi-Kamols and Tesnière's Dependency Grammars.* Proceedings of the 4th International Conference on Human Language Technologies — the Baltic Perspective, Frontiers in Artificial Intelligence and Applications, Vol. 219, IOS Press, pp. 233–240

Petr Pajas, Jan Štěpánek. 2006. *XML-Based Representation of Multi-Layered Annotation in the PDT 2.0.* Proceedings of the LREC Workshop on Merging and Layering Linguistic Information (LREC 2006), pp. 40–47.

Lucien Tesnière. 1959. *Éléments de syntaxe structurale.* Klincksieck, Paris. (Translation to Russian: Теньер Л. 1988. *Основы структурного синтаксиса.* Ред. В.Г. Гак. Москва, Прогресс.)

ISSN 1736-6305 Vol. 11
http://hdl.handle.net/10062/16955