Dependency-Based Hybrid Model of Syntactic Analysis for the Languages with a Rather Free Word Order

Guntis Bārzdiņš, Normunds Grūzītis, Gunta Nešpore and Baiba Saulīte
Institute of Mathematics and Computer Science
University of Latvia
Raiņa bulv. 29, Rīga, LV-1459, Latvia
guntis@latnet.lv, {normundsg,gunta,baiba}@ailab.lv

Abstract

Although phrase structure grammars have turned out to be a more popular approach for analysis and representation of the natural language syntactic structures, dependency grammars are often considered as being more appropriate for free word order languages. While building a parser for Latvian, a language with a rather free word order, we found (similarly to TIGER project for German and Talbanken05 for Swedish) that none of these models alone is adequate. Instead, we are proposing an original hybrid formalism that is strongly built on top of the dependency model borrowing the concept of a constituent from the phrase structure approach for representing analytical (multiword) forms. The proposed model has been implemented in an experimental parser and is being successfully applied for description of a wide coverage grammar for Latvian.

1 Introduction

The reported research is part of an interdisciplinary project[*] that aims to develop semantic resources and methodologies for automatic meaning extraction from Latvian texts. This ultimate goal requires the lower levels of the language analysis, namely, morphology, syntax and lexical semantics, to be properly understood and implemented in the first place. The advantage of our semantic framework is that we are not concerned with the full disambiguation at the level of parsing, as the final disambiguation can be, hopefully, postponed to the semantic processing layers involving frame semantics (like FrameNet) and ontologies (like SUMO) and reasoning techniques. Our experiences of building such syntax parser for Latvian via original hybrid model techniques are described in this paper.

Morphological analysis nowadays is a solved problem for virtually any language group. However, a deep and comprehensive analysis and representation of the syntactic structure of an arbitrary sentence, is still a challenge, illustrated by the wide variety of formalisms attempted in non-English treebanks, such as TIGER for German (Brants et. al., 2002) or Talbanken05 for Swedish (Nivre et. al., 2006). To name a few, difficulties are typically caused by discontinuous constituents, coordinate structures and analytical forms (like the ambiguous prepositional phrases (Volk, 2006)).

Latvian belongs to the Baltic language group — it is a highly inflective synthetic language with a rather free word order. We are using the term *rather* due to the fact that there is virtually no language with an absolutely free word order and vice versa (Saussure, 1966). We are claiming that Latvian has one of the most liberal word orderings. In terms of the grammar structure Latvian is closely related to Lithuanian and also to Slavonic languages (int. al. many Central European languages). Therefore the model we have developed and tested for Latvian might be of interest also for other languages.

[*] SemTi-Kamols project at the Institute of Mathematics and Computer Science (UL). Anno 2005. www.semti-kamols.lv

There are two mainstream approaches that are typically considered when developing a syntactically annotated corpus and a forthcoming parser — phrase structure (constituency) model or dependency model (Nivre, 2002). Although constituency and dependency grammars are at least weakly equivalent (Gaifman, 1965), i.e. mutually transformable, they suggest significantly different views and methodologies with their own respective advantages and disadvantages.

Parsers for languages with a rather strict word order typically follow a top-down approach: sentences are split into phrases or constituents, which are then split into more fine-grained constituents. Conventionally, formalization of constituents is done by means of a *phrase structure* (*constituency, generative*) *grammar* (Chomsky, 1957; Marcus et. al., 1993).

Languages with a rather free word order can be more naturally (with considerably smaller number of rules) described following the bottom-up approach: from the surface to the model by drawing subordination links that are connecting individual words. Conventionally, these links are formalized via a *dependency grammar* (Tesnière, 1959; Mel'čuk, 1988; Hajičová et. al., 2001).

However, in practice the argument of the word order has not been a very strong one. Phrase structure rather than dependency structure treebanks have turned out to be a more popular approach also for synthetic free word order languages, although additions like functional annotations there are often added (Nivre, 2002) or efforts are made to create both types of syntactically annotated corpora, e.g. (Nivre et. al., 2006). One of the phrase structure popularity reasons might be the compatibility in methods, algorithms and tools with the English-speaking community.

The choice of an annotation scheme in fact is not limited just to the one or the other candidate. Various hybrid models have been proposed also before, like the different versions of head-driven phrase structure grammars (HPSG) (Pollard and Sag, 1994) or the TIGER annotation scheme (Brants and Hansen, 2002) and its predecessor (Skut et. al., 1997). The latter one seems to be the most advanced approach towards a real hybrid model for syntactic analysis: a sentence there is represented as a graph whose nodes are constituents and edges — syntactic functions. This allows TIGER to adequately represent such phenomena as discontinuous constituents, which are typical for the free word order languages. However, to support languages with even more liberal word order than in the case of German, the TIGER model can be further empowered with more explicit dependency grammar elements as will be described in the sections 2 and 3, where we present our original hybrid approach. An initial evaluation of the approach is given in the section 4.

2 Our Hybrid Parsing Method

Our hybrid parsing method is strongly based on the pure dependency parsing mechanism described by Covington (2001; 2003). Meanwhile it is fundamentally extended with a constituency mechanism to handle analytical multi-word forms consisting of fixed order mandatory words. This enables us to elegantly overcome the limitation of the pure dependency grammars, where all dependants are optional and totally free-order. In our approach a head and a dependant don't have to be single orthographic words anymore.

The merging of the two approaches though is not straightforward — to do so we had to introduce a concept of "x-word", which in a sense is the core idea of our method. As will be seen in the further explanation, x-words are devices that cancel off substrings in parsing and they act as glue between the two worlds due to their dual nature:

- x-words can be viewed as non-terminal symbols in the phrase structure grammar, and as such during the parsing process substitute all entities forming respective constituents;
- the dependency parser treats x-words 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 concept of x-word, in fact, is analogous to the "nucleus" — the primitive element of syntactic description introduced by (Tesnière, 1959) and discussed in-depth and exploited in (Järvinen and Tapanainen, 1998).

It also bears some similarity to the "classical" HPSG approach (Pollard and Sag, 1994), where features of a phrase are handed over via the head of the phrase (i.e. a constituent as whole is represented only by the features of its head). The main difference of our model is that x-word is a new artificial word with artificial morphological properties inherited in the controlled way from all constituents that are forming the x-word.

In our approach all complex text structures with fixed word order, like prepositional phrases and analytical forms (perfect tenses) of a predicate, can be seen as (substituted by) x-words (see Figure 1). Section 3 provides a more detailed description of the intended x-word usage.

By iteratively substituting all analytical word forms in the text with the corresponding x-words, we are ending up with a simple sentence structure, which can be described and parsed by simple word-to-word (including x-word) dependencies. The only requirement thus is an agreement on the specified morphological features (as in Figure 1). Agreement is established via Prolog-style feature unification (Covington, 2003).

```
([_,[v,aux,Tense,Nr,_]],
 [_,[v,aux,past,Nr,_]],
 [_,[v,m,0,0,Trans]])→
[x-verb,[v,m,Tense,Nr,Trans,perf]]
```

Figure 1. A simplified example of an x-word declaration: substitution of an analytical form of a verb (like *ir bijis jādod* 'have had to give'). Constants are in lower case. Capitalized are variables that have to agree on values or are to be inherited.

For languages with a rather free word order constituents of analytic forms are required to appear in a fixed order, however, dependants of such constituents in general appear in a free order according to the rules of the dependency grammar and thus can interleave in between. The consequence is that x-words are defined only by their mandatory constituents while the optional ones (if any) are attached implicitly via the pure dependency grammar.

An illustration of a hybrid parse tree generated according to the described x-word based hybrid model is shown in Figure 3.

Despite its conceptual simplicity, the proposed method is very powerful and can be used to parse different phenomena (see section 3) in languages both with rather free or strict word order.

2.1 Implementation

In general, parsing of dependencies can be based on two simple tables (Covington, 1990):

- a list of word forms and their morphological descriptions (let us name it A-table);
- a list of possible head-dependent dependency pairs, declaring which word forms may be linked by which syntactical roles (let us name it B-table).

The parsing is reduced to the search problem for the parse tree satisfying these given constraints.

In our implementation an automatic acquisition of the table A is done on-the-fly by exploiting a morphological analyzer over the words of input sentence (see Figure 2 for an illustration of the resulting A-table).

Additionally to this infrastructure inherited from Covington (1990; 2003), we have introduced one more table (X-table), which is a list of complex, fixed word order patterns along with their x-word substitutions (as sketched in Figure 2). An x-word is composed via production rules analogous to those of the constituency grammar (only it is written in a bottom-up direction). The difference is that only the mandatory constituents of an x-word are explicitly declared, while their optional dependents are described by the regular dependency rules (B-table). X-words can be nested in other x-words as well — either directly like in a constituency grammar, or indirectly via dependency rules of B-table.

From the point of view of the B-table, simple word or x-word heads/dependants are treated equally.

A-Table	
Word	Morphological Features
vasarā	[n,f,sg,loc]
var	[v,aux,present,pl,trans]
peldēties	[v,m,inf,0,intrans]

X-Table		
x-Word	Morphology	Constituents
x-coord	...	...
x-prep	...	...
x-verb	...	...

B-Table		
Function	Head	Dependant
modifier	[_,{v,m}]	[_,{n,loc}]
subject	[x-verb,{v,m,Nr}]	[_,{n,Nr,nom}]
attribute	[_,{n}]	[_,{n,gen}]

Figure 2. Simplified illustration of the tables A, X and B. Notation {..} — unordered conditions.

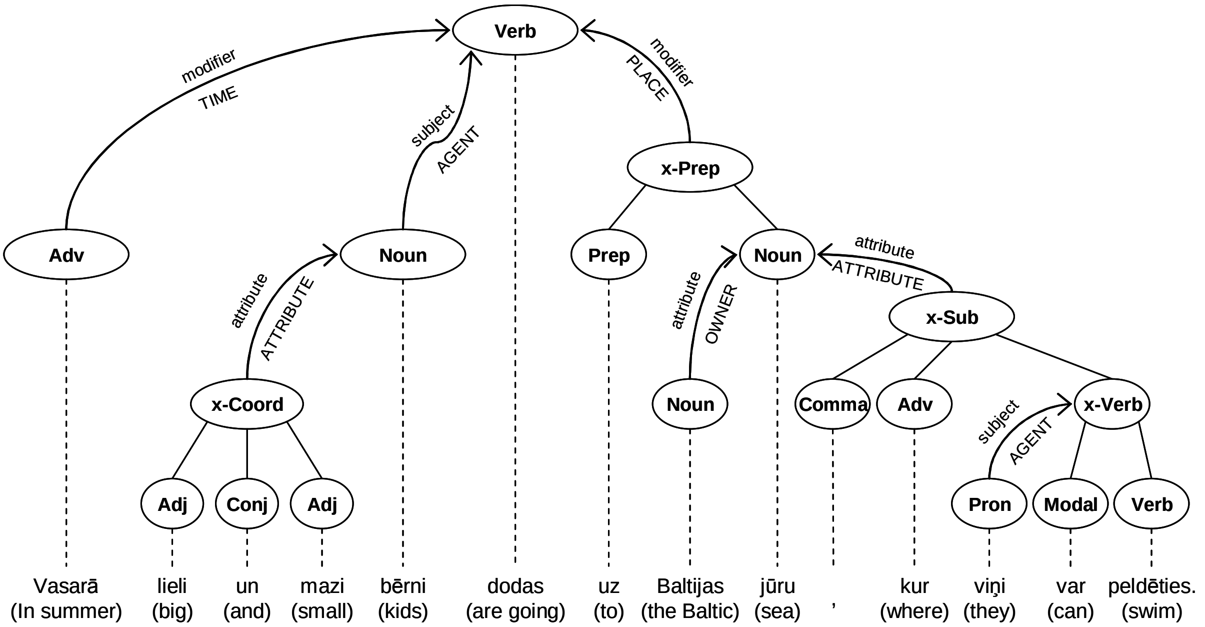


Figure 3. A shallow parse tree conforming to the hybrid model. Directed arcs stand for dependencies (optional), undirected — for constituents (mandatory). Nodes are words, either simple or complex (x-words).

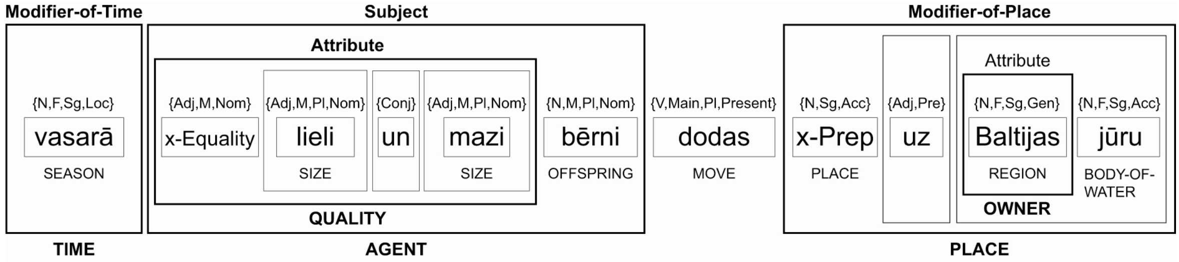


Figure 4. A chunk of the sentence presented in the Figure 3. Tree representation is encoded in the notation of the nested boxes.

Although an x-word as such in its adjacency is seen as syntactic primitive, its internal structure is parsed further as an independent subtree exploiting the fixed patterns and dependency connections defined in the X- and B-table respectively. Note that both explicit and implicit constituents interleave (e.g., 'UZ Baltijas JŪRU' in Figure 3).

To reduce parsing ambiguity, we have introduced one additional constraint in our parsing engine: each head is allowed to have only one dependant with the same syntactic role (function column in the B-table). For instance, this avoids more than one (uncoordinated) subject per predicate, which seems to be a natural constraint.

The proposal can be summarized as follows: we have added the mechanism of x-words to a combination of (Covington, 2003) + (Brants and Hansen, 2002). By introducing the x-words we have made the hybrid approach already proposed by the TIGER schema more straightforward and more powerful.

2.2 Visualization

Along with an original method of parsing, we have also introduced a space-saving graphical notation — nested boxes — in addition to the classical tree representation. In our notation each box corresponds to a single word (simple or complex x-word) and has both syntactic and semantic annotations. A list of morphological features and syntactic role is given at the top of a word/box; a label of an ontological concept and semantic role is given at the bottom (see Figure 4, illustrating the box-representation of the parse tree shown in Figure 3).

3 Methodology

In this section we will show how different well-known phenomena of syntactical analysis are handled by our hybrid parser.

3.1 Free Word Order

Considering analysis of a free word order, subordination relations are declared between parts of a simple sentence, assuming that each part basically is represented by a single word (see Figure 5). As a result, dependency grammar is defined by a set of head-dependent pairs, where only the agreement of morphological forms between both parts is significant, but not the order in which they appear in a sentence, since it doesn't have impact on the syntactic model.

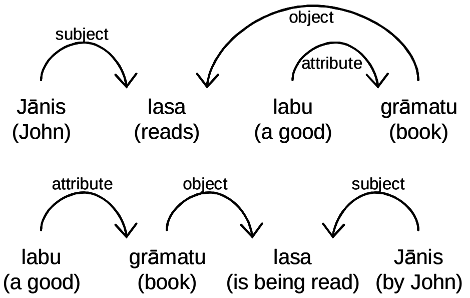
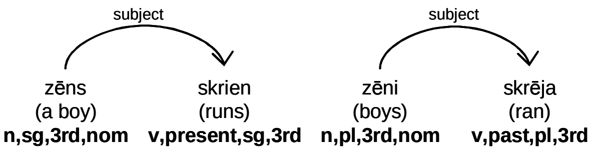


Figure 5. Dependency tree (arcs) remains the same for different readings of a sentence.

Out of the six possible subject-predicate-object orderings all the six are allowed in Latvian. Position of an adverb also is not constrained. Only attributes traditionally go before their heads.

3.2 Agreement

In Latvian as an inflective language agreement is very important phenomenon. It happens in both nominal (e.g. *liel<u>ā</u> māj<u>ā</u>* 'in a big house'), and verbal forms.



```
subj({n,Nr,Pers,nom},{v,Nr,Pers})
```

Figure 6. A single unification-based dependency rule will correctly accept all the subject(noun)-predicate pairs.

The head and the dependant of each dependency pair can be easily turned into constrained patterns (see Figure 6 for a simplified example), stating conditions on rich morphological features and inflectional agreement between both parts.

3.3 Constraints on the Left/Right Position

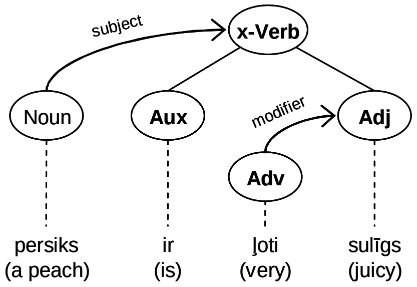
Apart form the internal structure of complex words, positional restrictions can be imposed on words per se. Although we are dealing with a language with a free word order in some cases the order of constituents is quite important. For example, in the already mentioned construction `attr([adj|_],[n|_])`, the constituents normally can not change their order.

The parser can be guided by an additional parameter of a dependency rule, indicating whether a head goes first or last against its dependant: `attr([adj|_],[n|_],right)`.

The fixed order of the words does not prevent them from being involved in other dependencies — they do not necessarily have to be placed together. For instance, the parser also accepts constructions like *liels koka galds* 'big wooden table'.

3.4 Analytical Forms of a Predicate

Rather free word order means that there exist rather strict constructions as well, i.e. analytical forms. The main part of a sentence that often is made up by few words in the same function is the predicate. We have described the following patterns of an analytical predicate in the X-table: perfect tenses, moods, passive voice, semantic modifiers (e.g. modal verbs), nominal and adverbial predicates.



```
([_,[v,aux,Tense,Nr,Prs]],
 [_,[adj,Gen,Nr,nom]]) →
[x-pred,[v,m,Tense,Nr,Prs,Gen,nom]]
```

Figure 7. A nominal predicate: auxiliary *to be* + an adjective. Modifier depends on the adjective.

Between the constituents of an analytical predicate other (dependant) parts of sentence may appear that is acceptable by the parser. In case of Latvian they are typically modifiers and attributes, which are related either to the predicate as whole or to a particular constituent (e.g., Figure 7). Note that such cases are also related to the phenomenon of discontinuous constituents (see section 3.7).

3.5 Prepositional Phrases

Prepositional phrases are regarded as x-words consisting of a preposition (or rarely — postposition) and a nomen in an appropriate (fixed) form. The nomen may be further involved as a root for a rich sub tree of dependants — all the structure will be regarded as a single x-word like in Figure 8.

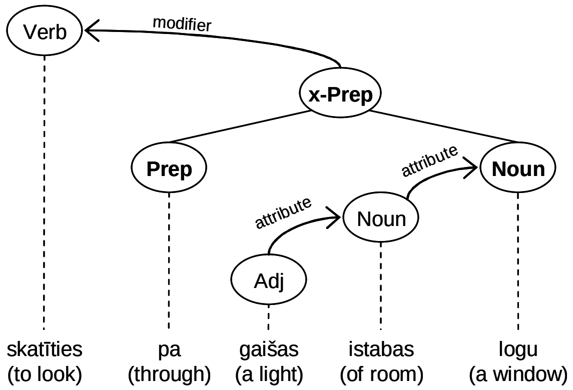


Figure 8. An x-word-driven prepositional phrase (*to look through a window of a light room*).

3.6 Coordinate Structures

Another well known issue concerns coordinate structures. The notion of an x-word can be clearly used to describe coordinated parts of a sentence as well (as illustrated in Figure 9).

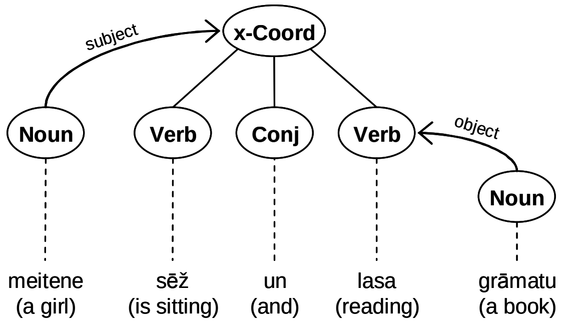


Figure 9. A typical pattern of a coordination structure that is parsed using the x-word mechanism. In this case the coordination results in a predicate.

Coordinated parts of sentence can be regarded as a single x-word, because syntactically they take the same position. Morphological features are in agreement, thus can be inherited with no loss of information.

3.7 Discontinuous Constituents

The widely discussed phenomenon of discontinuous constituents is one of the main issues if dealing with a phrase structure grammar. Dependency grammars on the contrary are not affected much by this problem — non-projective parse trees are very infrequent phenomena, since dependency grammars are not based on constituents and the root element of each parse tree is a verb (predicate) to which all the other syntactic primitives are connected, either directly or recursively via its dependants. Moreover, we are basically interested in texts where neutral word order prevails, i.e., in a written text but not in a speech. We also exclude from the scope of written texts some specific usages of a language, e.g. poetry.

In our approach discontinuous x-words are implicitly covered by the natural interleaving of dependants within the x-words (see sections 2.1 and 3.4). However, there is a limitation — dependants that linearly stand inside of an x-word are not allowed to be connected to the x-word as whole but to a particular constituent of it — which, in fact, is a semantically motivated restriction (at least for Latvian).

3.8 Subordinate and Coordinate Clauses

It is obvious that subordinate and coordinate clauses are based on a simple sentence structure. Therefore in our model subordinate clauses are seen as x-words as well — they link to the principal clause as a single part of a sentence (both syntactically and semantically), and typically they are dependants of a single word (simple or complex one). An example has been already shown in Figure 3. Thereby, an artificial part-of-speech must be introduced for a subordinate clause.

Clauses being in coordination relationship could be joined under an artificial node *sentence*, similarly as it is illustrated with coordinate structures in section 3.6. However, from the point of view of semantic structure each coordinated clause is treated as a separate sentence. Such an x-word would only introduce unnecessary ambiguity together with grammar patterns for coordinated verbs: by application of dependency rules the coordinated parts of sentence can be expanded up to coordinated clauses.

4 Evaluation

It should be noted that we are not considering any performance and algorithmic complexity aspects in the scope of this paper. Moreover, we would like to avoid any premature discussion on optimization or disambiguation to keep the model descriptive and clean until the stage of the semantic analysis.

The described hybrid parsing method has been implemented in a running parser of Latvian. Performance of the naïve and straightforward implementation is in the range of few seconds per sentence and is acceptable for verification purposes of the grammar.

The grammar is already able to recognize most types of frequent syntactic structures. If an arbitrary sentence can not be parsed successfully, it is mainly because of “routine” work needed to add the missing table entries to the system. However, it is feasible that a significant amount of work is still pending to accomplish a near-complete coverage.

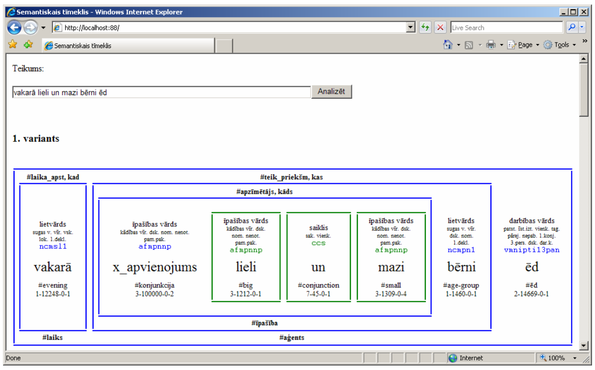
Currently we have formalized ~450 patterns of x-words (X-table) and ~200 dependency rules (B-table). A-table, as it was mentioned earlier, for each sentence is built on-the-fly by exploiting a morphological analyzer of Latvian. Although the number of patterns/rules is still small, part of them have been detected as overlapping, or are too general. This results in high number of ambiguities for the respective sentences. Due to this, in parallel we are developing an automated consistency checker to detect the possible inconsistencies or overlapping in the hand-crafted rules.

On the other hand, free word order structures by default are more ambiguous than the corresponding analytical constructions. Therefore, we produce all the possible parse trees for each sentence and consider the result correct and sufficient for the further semantic parsing stage, if all these trees are syntactically correct and the semantically correct tree is among them. We agree with (Tesnière, 1959) that the syntactic structure follows from the semantic structure. Therefore we regard disambiguation as a separate problem and in the current stage of analysis we only do care that there are syntactically valid trees produced.

Some constructions are not implemented in the parser yet (e.g. semi-predicative components and participial phrases), but we believe that there are no principal problems in dealing with these constructions.

A screenshot of a running application is given in Figure 11. Although the model and the parser were made taking into account the Latvian language only, the parser that is based on the three clear-cut tables has turned out to be language independent.

One might ask why we haven’t tried to extract the grammar from a treebank. It has been shown that if there is a sufficiently large treebank available (at least about 20 000 manually annotated sentences), it is possible to learn the grammar at a certain extent from the treebank (Charniak, 1996). Unfortunately there is no large scale Latvian treebank available. Actually, there is no publicly accessible treebank at all. Moreover, the corpus has to be annotated with a grammar of interest. Instead we are planning to develop an experimental treebank on the basis of the approach and the parser presented.



Teikums:

vakarā lieli un mazi bērni ēd

Analizēt

1. variants

Figure 11. A screenshot of the user-interface of the experimental Latvian syntax parser. It is implemented in SWI-Prolog with a web-browser front-end.

5 Conclusion

We have experimentally verified that the proposed hybrid model, which is strongly based on the dependency grammar approach, can be used to describe languages both with rather free or strict word order. Even if the computational performance and simplicity is better for phrase-structure grammars, the construction of a wide coverage grammar might be more convenient via a layer of the proposed hybrid approach. Straightforward compatibility between the syntactic and semantic structures in case of the dependency grammar is also of a great importance.

In order to adapt the parser for other languages "only" the three tables (A, X and B) have to be produced describing morphology and syntax of the particular language.

Acknowledgements

Project is funded by the National Research Program in Information Technologies and is partially supported by European Social Fund. Also we thank our colleagues and reviewers of this paper for their valuable comments and references.

References

Sabine Brants and Silvia Hansen. 2002. *Developments in the TIGER Annotation Scheme and their Realization in the Corpus*. In Proceedings of the Third Conference on Language Resources and Evaluation (LREC 2002), pp. 1643–1649

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius and George Smith. 2002. *The TIGER Treebank*. In Proceedings of the Workshop on Treebanks and Linguistic Theories

Eugene Charniak. 1996. *Tree-Bank Grammars*. In AAAI/IAAI, Vol. 2, pp. 1031–1036

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

Michael A. Covington. 1990. *A Dependency Parser for Variable-Word-Order Languages*. Research Report AI-1990-01, Artificial Intelligence Center, The University of Georgia

Michael A. Covington. 2001. *A fundamental Algorithm for Dependency Parsing.* In Proceedings of the 39th Annual ACM Southeast Conference. Eds. John A. Miller and Jeffrey W. Smith, pp. 95–102

Michael A. Covington. 2003. *A Free-Word-Order Dependency Parser in Prolog*. Prolog Natural Language Tools, The University of Georgia http://www.ai.uga.edu/mc/dparser/dparser.pdf

Haim Gaifman. 1965. Dependency Systems and Phrase-Structure Systems. *Information and Control*, 8:304–307

Eva Hajičová, Jan Hajič, Martin Holub, Petr Pajas, Veronika Kolářová-Řezníčková, Petr Sgall, Barbora Vidová Hladká. 2001. *The Current Status of the Prague Dependency Treebank*. In Proceedings of the 5th International Conference on Text, Speech and Dialogue, Železná Ruda-Špičák, Czech Republic, Springer-Verlag Berlin Heidelberg New York, pp. 11–20

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

Igor Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. Albany, N.Y.: The State University of New York Press

Mitchell P. Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330

Joakim Nivre. 2002. *What kinds of trees grow in Swedish soil? A comparison of four annotation schemes for Swedish*. In Hinrichs, E. and Simov, K. (eds) Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT2002)

Joakim Nivre, Jens Nilsson and Johan Hall. 2006. *Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation*. In Proceedings of LREC, pp. 1392–1395

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago: University of Chicago Press

Ferdinand de Saussure. 1966. *Course in General Linguistics*. New York: McGraw-Hill Book Company

Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Klincksieck, Paris

Martin Volk. 2006. *How bad is the problem of PP-attachment? A comparison of English, German and Swedish*. In Proceedings of ACL-SIGSEM Workshop on Prepositions, Trento

Wojciech Skut, Brigitte Krenn, Thorsten Brants and Hans Uszkoreit. 1997. *An Annotation Scheme for Free Word Order Languages*. In Proceedings of the Fifth Conference on Applied Language Processing (ANLP 1997) pp. 27–28