Problems of Inducing Large Coverage Constraint-Based Dependency Grammar

Ondřej Bojar
Institute of Formal and Applied Linguistics, ÚFAL MFF UK
Malostranské náměstí 25
CZ-118 00 Praha 1
Czech Republic
obo@cuni.cz

Abstract

This article describes an attempt to implement constraint-based dependency grammar for Czech, a language with rich morphology and free word order, in the formalism Extensible Dependency Grammar (XDG). The grammar rules are automatically inferred from the Prague Dependency Treebank (PDT) and constrain dependency relations, modification frames and word order, including non-projectivity. Although these simple constraints are adequate from the theoretical point of view, their combination is still too weak and allows an exponential number of solutions for a sentence of n words.

1 Motivation

Current approaches to syntactic analysis of Czech and other languages with freer word order have limitations that are important from the theoretical point of view. First, all the available parsers are restricted to the surface syntactic analysis and there is no simple of extending it to include deep syntactic (for instance tectogrammatical) level of representation. Second, the available statistical parsers produce only one solution for a given sentence, ignoring the possibility of the syntactic ambiguity of a sentence. And last but not least, the available parsers[1] are by nature statistical and do not contribute to the explanation of syntactic phenomena very much.

Several declarative (relational) approaches to syntax analysis overcoming these problems are available, including well known formalisms such as HPSG or LFG, or the robust constraint-based dependency parsing by Foth et al. (2004). Another promising formalism is the Extensible Dependency Grammar (XDG, Debusmann et al. (2004)). None of these approaches has ever been tested on a language with rich morphology and freer word order in a large scale.

With respect to ongoing research within the theoretical framework of Functional Generative Description (Sgall et al., 1986), the Extensible Dependency Grammar is a formalism that excellently fits our needs:

- XDG is dependency based, as FGD is.
- XDG distinguishes between immediate dominance (ID, dependency) relations and linear precedence (LP) restrictions; constraints are allowed to speak about these two dimension separately as well as simultaneously and the dimensions are mutually constraining each other. It is easy to handle non-projective constructions in XDG. Both these issues are important with respect to the relatively free word order of Czech.
- XDG allows for adding new dimensions of language description, such as the deep syntactic (tectogrammatical) level. FGD's main objective is deep syntactic structure.
- XDG effectively works with ambiguity: morphological, syntactic and lexical ambiguity during parsing is stored in a compact underspecified form as long as possible. The multiplication of orthogonal options is postponed until actually needed.

The task of implementing a large coverage grammar with XDG is interesting for another reason, too. Up to now, only small scale grammars have been implemented in XDG. These grammars illustrated efficient and elegant treatment of various complex syntax and semantic phenomena in XDG (Duchier and Debusmann, 2001; Debusmann and Duchier, 2003). However, the grammars were always tailored to a few test sentences and constraints implemented in XDG never had to cope with syntactic ambiguity of a grammar inferred from

[1] Rare exceptions include an unpublished parser for Czech by Zdeněk Žabokrtský.

a larger amount of data. There are excellent data sources for Czech language from which such a large scale grammar can be collected: the Prague Dependency Treebank (PDT[2]) and the Czech valency lexicon (Vallex, Žabokrtský et al. (2002)).

2 Introduction

2.1 Properties of Czech Language

Table 1 summarises some of the well known properties of Czech language[3]. Czech is an example of Slavonic languages. It is an inflective language with rich morphology and relatively free word order allowing non-projective constructions. However, there are important word order phenomena restricting the freeness. One of the most prominent examples are clitics, particles that occupy a very specific position within the whole clause. The position of clitics is very rigid and global within the sentence. Locally rigid is the structure of (non-recursive) prepositional phrases or coordination. Other elements, such as the predicate, subject, objects or other modifications may be nearly arbitrarily permuted.

Moreover, like other languages with freer word order, Czech allows non-projective constructions (crossing dependencies). Only about 2% of edges in the PDT are non-projective, but this is enough to make nearly a quarter (23.3%) of all the sentences non-projective.

The task of parsing languages with relatively free word order is much more difficult than parsing of English, for example, and new approaches still have to be searched for. Rich morphology is a factor that makes parsing more time and data demanding.

2.2 Overview of the Intended Multi-dimensional Czech Dependency Grammar

Figure 1 summarises data sources available for a Czech grammar induction. The PDT contains surface syntactic (analytic, AT) as well as deep syntactic (tectogrammatical, TG) sentence annotations. The Czech valency lexicon is under development, and alternatively, the valency lexicon collected while annotating the tectogrammatical level of PDT could be used.

	Czech	English
Morphology	rich	limited
	≥ 4,000 tags	50 used
	≥ 1,400 actually seen	
Word order	free with rigid global phenomena	rigid
Known parsing results		
Edge accuracy	69.2–82.5%	91%
Sentence correctness	15.0–30.9%	not reported

Table 1: Properties of Czech and English.

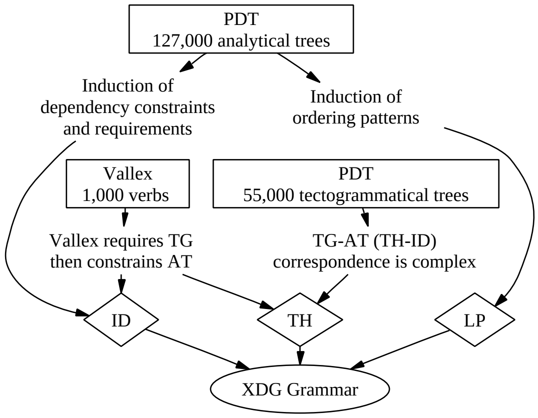
PDT
127,000 analytical trees
Induction of dependency constraints and requirements
Induction of ordering patterns
Vallex
1,000 verbs
PDT
55,000 tectogrammatical trees
Vallex requires TG then constrains AT
TG-AT (TH-ID) correspondence is complex
ID
TH
LP
XDG Grammar

Figure 1: Czech data sources available for XDG grammar.

A grammar in the formalism of XDG could be inferred from these sources addressing the immediate dominance (ID), linear precedence (LP) and thematic (TH, predicate-argument) dimensions.

Only a part of this overall picture has been implemented so far. First, the correspondence between tectogrammatical and analytic levels is quite complicated, some nodes have to be deleted, some nodes have to be added. Second, the tectogrammatical valency information from Vallex is mostly useful only if a tectogrammatical structure is considered, only then the constraints addressing surface realization can be fully exploited. Therefore, in the first approach the current grammar implementation focuses only on ID an LP levels.

3 Description of the Grammar Parts

The experimental XDG grammar induced from the PDT utilizes basic principles that are linguistically motivated and traditionally used in many varieties of dependency grammars, in-

[2] See Hajič et al. (2001) and Hajičová et al. (2000).

[3] Data by Collins et al. (1999), Holan (2003), Zeman (http://ckl.mff.cuni.cz/˜zeman/projekty/neproj) and Bojar (2003). Consult Kruijff (2003) for measuring word order freeness.

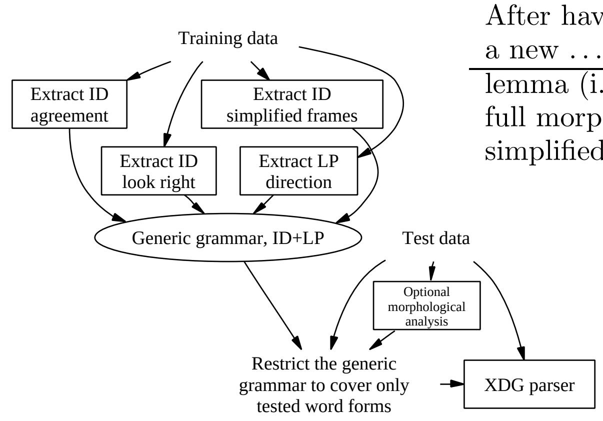
Training data
Extract ID agreement
Extract ID simplified frames
Extract ID look right
Extract LP direction
Generic grammar, ID+LP
Test data
Optional morphological analysis
Restrict the generic grammar to cover only tested word forms
XDG parser

Figure 2: XDG grammar parts and evaluation.

cluding XDG. The current XDG grammar extracted from the PDT consists of the following parts: ID Agreement, LP Direction, ID Simplified Frames and ID Look Right. For every part independently, the properties of individual lexical entries (with an arbitrary level of lexicalization) are collected from the training data. The contributions are then combined into XDG lexical entries and classes in a conjunction manner: when parsing, every input word must match one of the observed configurations in all the grammar parts.

For practical reasons (memory and time requirements), the grammar finally used in the XDG parser is restricted to the word forms of the test sentences only. Figure 2 summarizes the pipeline of grammar extraction and evaluation.

3.1 Grammar Non-lexicalized in General

XDG is designed as a lexicalized formalism, most syntactic information is expected to come from the lexicon. Conversely, to make the most use of this approach, the information in an XDG grammar should be as lexicalized as possible.

Despite the size of the PDT (1.5 million tokens), there is not enough data to collect syntactic information for individual word forms and even lemmas.

All the grammar parts described below are therefore based on simplified morphological tags only (part and subpart of speech, case, number and gender). Table 2 justifies this simplification. Theoretically, full morphological tags could be used, but we would face sparse data problem if pairs or n-tuples of tags were examined.

After having observed	20,000	75,000	training sentences
a new ... comes every			test sentences
lemma (i.e. word)	1.6	1.8	test sentences
full morphological	110	290	test sentences
simplified tag	280	870	test sentences

Table 2: Lack of training data in PDT for full lexicalization.

3.2 ID Agreement

The ID Agreement part of the grammar allows for a specific edge type between a father and a daughter. The edge type is cross checked in both directions: from father and from the daughter.

Technically, the lexical entry of a father (with known morphological properties) contains a mapping from edge labels to morphological requirements on any possible daughter. If a daughter is connected via a particular edge label to this father, the daughter's morphology must match at least one of the requirements. Conversely, the daughter's lexical entry contains a mapping to restrict morphology of the father.

This approach helps to reduce morphological ambiguity of nodes: For every node, only such morphological analyses remain allowed which fit the intersection of requirements of all the connected nodes. During parsing, the ambiguous morphology of the node is reduced step by step, as more and more edges are assigned.

3.3 LP Direction

The LP Edge Direction part describes simplified linear precedence rules and handles non-projectivity. In the original design of XDG grammars, motivated by German, the LP dimension is used to describe *topological fields* (Bech, 1955). Unfortunately, the word order of Czech and other Slavonic languages does not exhibit similar word order restrictions in general. (To a very limited extent, one could think about three fields in a clause: preclitic, clitic and postclitic field.) However, there is often an important distinction between dependencies to the left and dependencies to the right.

Technically, every father at the LP dimension offers three fields: the left field of unlimited cardinality[4], head field to contain only the father

[4] In other words, unlimited number of outgoing LP edges can have the label LEFT and all edges labelled LEFT must be present first in the left-to-right ordering of nodes.

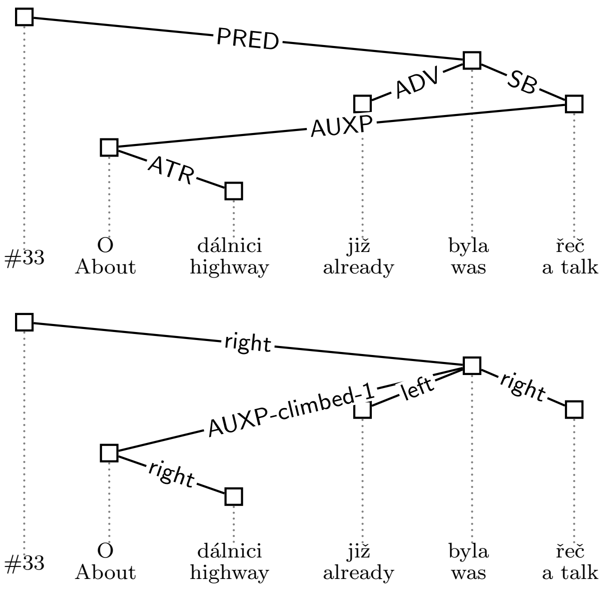

Figure 3: LP dimension to handle edge direction and non-projectivity.

itself and the right field of unlimited cardinality. The restriction on specific edge types allowed in a specific direction is handled by another principle: given a daughter connected to the father with an edge of a particular label at the ID dimension, the corresponding LP edge is allowed to have only some of the labels. As illustrated in Figure 3, under the preposition *about*, an (ID) edge labelled ATR can go to the right only, so the corresponding LP edge must have the label RIGHT.

Non-projectivity is forbidden in general but allowed for observed cases. This constraint is expressed in the LP tree while the ID tree is allowed to be non-projective in general. The LP tree is required to be projective and the exceptions are handled by the so-called *climbing principle*. In order to obtain a projective LP tree from a non-projective one, the tree is "flattened" by climbing. For example, the AUXP edge is non-projective in the ID tree in Figure 3. Moving the corresponding LP edge one step up from the governor *talk* to the governor *was*, the LP edge becomes projective.

To distinguish LP edges that had to climb from LP edges directly corresponding to ID edges, a set of extra LP labels is introduced: AUXP-CLIMBED-1, ATR-CLIMBED-1... The nodes where a climbed edge may land offer not just the left, head and right fields, but also the required amount of specific *-CLIMBED-* edges. There is no restriction on mutual linear ordering of the LEFT/RIGHT and *-CLIMBED-* edges.

The current model still lacks restrictive power to control the clitic position. Similarly, coordination is not modelled properly yet, because the cardinality of left and right fields is unrestricted in general (for example, both members of a coordination are allowed to appear on the same side of the conjunction). More adequate handling of these phenomena remains open for further research.

3.4 ID Simplified Frames

One of the crucial principles restricting available sentence analyses in XDG is the valency principle: Every father node allows only specific combinations and cardinalities of outgoing (ID) edges.

The ID Simplified Valency Frames ensure that a word doesn't accept implausible combinations of modifiers. Rarely, they ensure that a word has all its "modification requirements" saturated, because most of the modifiers are deletable anyway.

Current approaches[5] aim at distinguishing *complements* vs. *adjuncts*, i.e. modifications that are required vs. optional. However, there is no use of this distinction, if *deletability* of modifications is taken into account (in real Czech sentences, complements are often omitted). Any consistent grammar must reflect this optionality of complements.

The restrictive power of valency frames in XDG should therefore come from *interdependencies* of modifications (e.g. if a secondary object or a specific type of adjunct was observed, primary object must be present). The set of allowed combinations and cardinalities must be explicitly enumerated in the current XDG implementation. Future versions of this principle might accept a constraint network (for example a set of implications) of interdependencies.

To my knowledge, no published approach aims at discovering such interdependencies of particular modifications so far. On the other hand, there are too many unique frames observed under a given node type, so it is impossible to enumerate all of them.[6]

[5] See Sarkar and Zeman (2000) for comparison and references.

[6] Enumerating all seen modification frames would face a severe sparse data problem anyway as the number of unique modification frames steadily grows. In 81,000 sentences, there were 89,000 unique frames observed when describing the frames as lists of simplified tags of

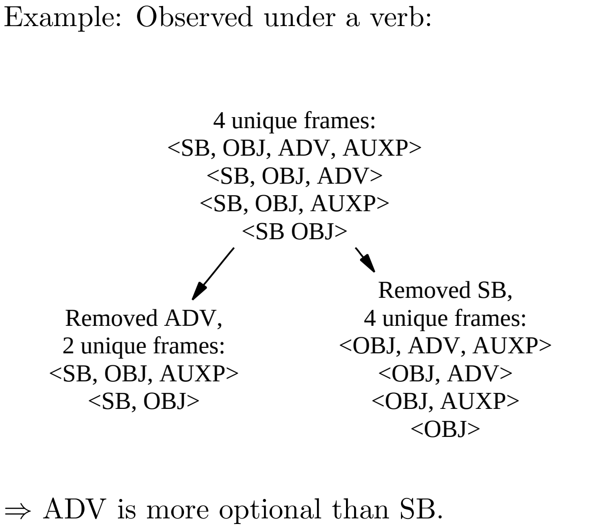

Figure 4: Identifying optional modifications in order to simplify the set of allowed modification frames.

Therefore, I implemented a naive algorithm to infer simplified modification frames: this algorithm automatically simplifies treatment of adjuncts and stores the complexity of interdependencies of other modifications by enumerating them. As sketched in Figure 4, the set of observed modification frames of a specific word class can be simplified by removing different modification types. When adverbial is removed under a verb, the set of modification frames shrinks to a half in size. When subject is removed instead, the set does not shrink at all. This indicates, that an adverbial has no effect on interdependencies of other modifications: an adverbial may be present or may not–half of the frames was observed with an adverbial, half of the frames had no adverbial.

This simplification is applied iteratively, until the number of unique frames is acceptable. The removed modifications are added to all the frames as optional.

A short example in Figure 5 illustrates the optionality order of modifications observed under (very few) verbs in present tense (POS=V, SUBPOS=B). The most optional modification (AUXP[7], a prepositional phrase) is torn off in the first step. The second torn-off modification is an adverbial (ADV) yielding simplified set of modification frames with 36 different frames.

It should be noted that the described solution is by no means a final one. The tasks of inducing modification frames and employing the frames to constrain syntactic analysis are very complex and deserve much deeper research.

Unique observed modification frames: 67
Set sizes when removing specific modifiers:
AUXP(50), ADV(50), OBJ(53), SB(55), AUXT(59), AUXG(60), PNOM(61), COORD(62), AUXR(62), AUXY(63), AUXX(65), AUXC(65), APOS(66), HEAD(67), EXD_PA(67), EXD(67)
Cumulative simplification:
67→(AUXP)→50→(ADV)→36...

Figure 5: Simplifying modifications of verbs in present tense.

3.5 ID Look Right

The generally accepted idea of dependency analysis is that head-daughter dependencies model syntactic analysis best. Dubey and Keller (2003) doubt this assumption and document that for German sister-sister dependencies (lexicalized case) are more informative.

Context		Neighbours		Sisters	
used	Head	Left	Right	Left	Right
Entropy	0.65	1.20	1.08	1.14	1.15

Table 3: Difficulty of predicting edge label based on simplified tag of a node and a node from close context.

Table 3 gives an indication for Czech: if the structure was already assigned, choosing the edge label is easiest when looking at morphological properties of the node and its head (lowest entropy). Contrary to Dubey and Keller, Czech with a very strong tendency for grammatical agreement confirms the generally accepted notion.

The ID Agreement principle is crucial in Czech and it is already employed in the grammar. Table 3 indicates also which context gives the second best hint: the right neighbour, i.e. the following word. Therefore, a new principle was added: ID Look Right: An incoming ID edge to a word must be allowed by the word class of its right neighbour.

The differences among sisters' and neighbours' contributions to the prediction of edge label are not very significant, so adding more constraints of this kind is still under consideration.

4 Results

To evaluate the grammar, only the first fixed point in constraint solving is searched. Given

all the daughters of a node.

[7] See Hajič et al. (2001) for explanation of the labels.

a sentence, XDG parser propagates all relevant and applicable constraints to reduce the number of analyses and returns an underspecified solution: some nodes may have unambiguously found a governor, for some nodes, several structural assignments may still remain applicable. At the first fixed point, none of the constraints can be used to tell anything more[8].

Two grammars were evaluated: first a version without the Look Right principle, second a version that included the new principle, too. The grammars were trained on sentences from the training part of the PDT and evaluated on 1,800 to 2,000 unseen sentences from the standard evaluation part of the PDT (devtest). The results are displayed in Table 4.

Note that the number of training sentences was relatively low (around 2 to 5% of the PDT), which explains the relatively high number of unsolved sentences (around 10 to 20%). Wider coverage of the grammar can be easily achieved by training on more data, but this would lead to significant growth of the number of solutions available. As indicated in the row Avg. ambiguity/node, a node has 8 to 9 possible governors (regardless the edge label). Compared with the average sentence length of 17.3 words, the grammar reduces the theoretically possible number of structural configurations to a half. At the first fixed point, the parser has enough information to establish only 3 to 5% of edges, an edge with a label can be assigned only to 2 to 4% of nodes. Out of the assigned structural edges, around 82% is correct, out of the assigned labelled edges, around 85% is correct. Again, training on more data should lead to a slightly lower error rate, but significantly less edges securely established, as confirmed by our results.

Contrary to our expectations, the adding the new principle Look Right did not help the analysis. The average ambiguity per node became even higher. There were slightly more edges securely assigned, but the correctness of this assignment has dropped.

Training sentences	2500	5000
		Unsolved sentences
Without Look Right	21.1	11.9
With Look Right	25.6	15.4
		Avg. ambiguity/node
Without Look Right	8.09	8.91
With Look Right	8.17	9.05
		Assigned structural edges
Without Look Right	4.4	3.3
With Look Right	4.7	3.5
		Correct structural edges
Without Look Right	82.3	82.5
With Look Right	81.9	81.0
		Assigned labelled edges
Without Look Right	3.4	2.3
With Look Right	3.6	2.5
		Correctly labelled edges
Without Look Right	85.9	85.9
With Look Right	85.0	83.5

Table 4: Results of underspecified solutions.

5 Discussion and Further Research

The presented results indicate several weak points in the described approach to constraint-based dependency parsing. All these points remain open for further research.

First, the tested version of XDG parser could not make any use of frequency information contained in the PDT.[9] Dienes et al. (2003) attempt at guiding the XDG parser by frequency information, but the research is still in progress. A similar constraint-based dependency parsing by Heinecke et al. (1998) inherently includes weight of constraints, but no directly comparable results were presented so far. (Foth et al. (2004) report edge accuracy of 96.63% on a corpus of 200 sentences with average length 8.8 words.)

Second, the current grammar relies on very few types of constraints. More constraints of different kinds have to be added to achieve better propagation. A related problem is the locality of the constraints. All the current constraints rely on a too local context. There are too many analyses available, because the local constraints are not powerful enough to check invariant properties of clauses or sentences as a whole.

Third, there are several kinds of expressions

[8] At fixed points, also called choice points, the constraint solver of the underlying system Mozart-Oz makes an arbitrary decision for one of the still underspecified variables and starts propagating constraints again. Another fixed points are reached and eventually a fully specified solution can be printed. Different solutions are obtained by making different decisions at the fixed points. The parser can be instructed to perform a complete search, but in our case there is no point in enumerating so many available solutions.

[9] In an experiment, frequency information was used as a threshold to ignore rare edge assignments. The thresholding resulted in lower coverage *and* lower precision.

that in fact have no dependency structure, such as names, dates and other multi-word expressions. The "dependency" analysis of such expressions in the PDT reflects more the annotation guidelines than some linguistic motivation. Separate treatment of these expressions by means of a sub-grammar would definitely improve the overall accuracy.

6 Conclusion

I described an experiment with constraint based dependency parsing of a language with rich morphology and freer word order. Although the constraints are linguistically adequate and serve well when employed on small-scale corpora, they face a serious problem when trained on large data sets. The constraints are too local and weak in order to restrict the number of available solutions.

To amend this problem, new kinds of constraints have to be developed. In order to achieve a plausible solution quickly, some sort of probabilistic guidance must be added to the constraint solver.

7 Acknowledgement

I'm grateful to Ralph Debusmann for his explanatory and immediate implementation support of new features needed in the XDG parsing system for this experiment. The work could not have been performed without the support of Programming Systems Lab headed by Gert Smolka (Universität des Saarlandes) and without the insightful guidance by Geert-Jan Kruijff and Denys Duchier.

References

Gunnar Bech. 1955. *Studien über das deutsche Verbum infinitum.* 2nd unrevised edition published 1983 by Max Niemeyer Verlag, Tübingen (Linguistische Arbeiten 139).

Ondřej Bojar. 2003. Towards Automatic Extraction of Verb Frames. *Prague Bulletin of Mathematical Linguistics*, (79–80):101–120.

Michael Collins, Jan Hajič, Eric Brill, Lance Ramshaw, and Christoph Tillmann. 1999. A Statistical Parser of Czech. In *Proceedings of 37th ACL Conference*, pages 505–512, University of Maryland, College Park, USA.

Ralph Debusmann and Denys Duchier. 2003. A meta-grammatical framework for dependency grammar.

Ralph Debusmann, Denys Duchier, Alexander Koller, Marco Kuhlmann, Gert Smolka, and Stefan Thater. 2004. A relational syntax-semantics interface based on dependency grammar. Technical report. Available at `http://www.ps.uni-sb.de/Papers`.

Péter Dienes, Alexander Koller, and Marco Kuhlmann. 2003. Statistical a-star dependency parsing. In Denys Duchier, editor, *Prospects and Advances of the Syntax/Semantics Interface*, pages 85–89, Nancy.

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103, Sapporo.

Denys Duchier and Ralph Debusmann. 2001. Topological dependency trees: A constraint-based account of linear precedence. In *39th Annual Meeting of the Association for Computational Linguistics (ACL 2001).*

Kilian Foth, Wolfgang Menzel, and Ingo Schröder. 2004. Robust parsing with weighted constraints. *Natural Language Engineering.* in press.

Jan Hajič, Jarmila Panevová, Eva Buráňová, Zdeňka Urešová, and Alla Bémová. 2001. A Manual for Analytic Layer Tagging of the Prague Dependency Treebank. Technical Report TR-2001-, ÚFAL MFF UK, Prague, Czech Republic. English translation of the original Czech version.

Eva Hajičová, Jarmila Panevová, and Petr Sgall. 2000. A Manual for Tectogrammatic Tagging of the Prague Dependency Treebank. Technical Report TR-2000-09, ÚFAL MFF UK, Prague, Czech Republic. In Czech.

Johannes Heinecke, Jürgen Kunze, Wolfgang Menzel, and Ingo Schöder. 1998. Eliminative parsing with graded constraints. In *Proceedings of COLING-ACL Conference*, Montreal, Canada.

Tomáš Holan. 2003. K syntaktické analýze českých(!) vět. In *MIS 2003*. MATFYZPRESS, January 18–25, 2003.

Geert-Jan M. Kruijff. 2003. 3-phase grammar learning. In *Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development.*

Anoop Sarkar and Daniel Zeman. 2000. Automatic Extraction of Subcategorization Frames for Czech. In *Proceedings of the 18th International Conference on Computational Linguistics (Coling 2000)*, Saarbrücken, Germany. Universität des Saarlandes.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects.* Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands.

Zdeněk Žabokrtský, Václava Benešová, Markéta Lopatková, and Karolina Skwarská. 2002. Tektogramaticky anotovaný valenční slovník českých sloves. Technical Report TR-2002-15, ÚFAL/CKL, Prague, Czech Republic.