

Problems of Inducing Large Coverage Constraint-Based Dependency Grammar for Czech

Ondřej Bojar

Center for Computational Linguistics, MFF UK
Malostranské náměstí 25, CZ-118 00 Praha 1, Czech Republic
obo@cuni.cz

Abstract. This article describes an attempt to implement a 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 linguistic point of view, their combination is still too weak and allows an exponential number of solutions for a sentence of n words.

1 Introduction

Czech is a thoroughly studied Slavonic language with extensive language data resources available. Traditionally, most of the research on Czech is performed within the framework of Functional Generative Description (FGD, [1]). This dependency-based formalism defines both surface syntactic (*analytic*) and deep syntactic (*tectogrammatical*, syntactico-semantic) level of language description. Language data sources for Czech include Prague Dependency Treebank (PDT, [2, 3]) and Czech valency lexicon (VALLEX, [4]). Specific properties of Slavonic languages and Czech in particular make the task of syntactic analysis significantly more difficult than parsing English. Available parsers of Czech ([5], [6] and an adapted version of [7]) are statistical, aimed at surface syntactic analysis and there is no simple way to extending them to include deep syntactic analysis. Up to now, no attempt has been made to approach large-coverage syntactic analysis with a constraint-based technology.

Extensible Dependency Grammar (XDG, [8]) is a promising relational framework aimed at multi-dimensional constraint-based analysis of languages. So far, 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 [9, 10]. 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 a larger amount of data.

This paper describes a first experiment of inducing a large-coverage XDG grammar for Czech from PDT.¹

¹ A more detailed description is given in [11].

1.1 Properties of Czech Language

Table 1 summarises some of the well known properties of Czech language². Czech 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 freedom. One of the most prominent examples are clitics, i.e. pronouns and 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. Such permutations correspond to the topic-focus articulation of the sentence. Formally, the topic-focus articulation is described at the deep syntactic level.

Moreover, like other languages with relatively free word order, Czech allows non-projective constructions (crossing dependencies). Only about 2% of edges in 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.

	Czech	English
Morphology	rich ≥ 4,000 tags ≥ 1,400 actually seen	limited 50 used
Word order	free with rigid global phenomena	rigid
Known parsing results		
Edge accuracy	69.2–82.5%	91%
Sentence correctness	15.0–30.9%	43%

Table 1. Properties of Czech compared to English.

1.2 Overview of the Intended Multi-dimensional Czech Dependency Grammar

Figure 1 summarises data sources available for a Czech grammar induction. PDT contains surface syntactic (analytic, AT) as well as deep syntactic (tectogrammatical, TG) sentence annotations. On the surface level, every input token (words and punctuation marks) is represented as a node in the ordered

² Data by [5], [12], Zeman (<http://ckl.mff.cuni.cz/~zeman/projekty/neprojl>), [13] and [14]. Consult [15] for measuring word order freeness.

rooted (dependency) tree. The arcs in the analytic tree are labelled with syntactic roles of the words such as Sb, Obj, Adv. See the top tree on Figure 3 below for an illustration.

The tectogrammatical level represents the predicate-argument structure of the sentence as well as some other linguistic phenomena such as information structure and coreference. Only autosemantic words are represented as nodes in the tectogrammatical tree and some extra nodes have to be added in order to saturate the predicate with all its arguments. (The arguments are known either from the context or from the world knowledge.) The knowledge about what kinds of arguments are required by the predicate comes from a valency lexicon. The Czech valency lexicon VALLEX is under development, and alternatively, the valency lexicon collected while annotating the tectogrammatical level of PDT could be used.

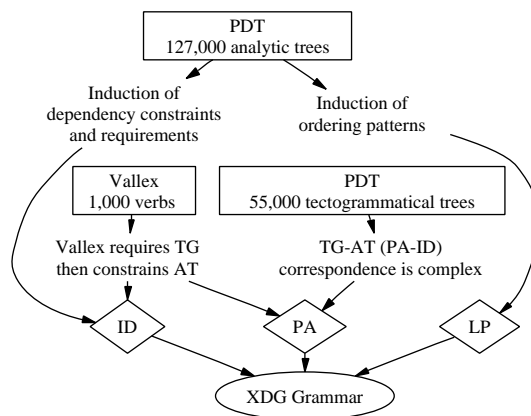


Fig. 1. Czech data sources available for XDG grammar.

A grammar in the formalism of XDG could be inferred from these sources addressing the immediate dominance (ID), linear precedence (LP) and predicate-argument (PA) dimensions. In XDG, the ID dimension is used to represent surface syntactic dependency structure of the sentence. The LP dimension is used to express constraints of word order that must be fulfilled in a correct parse. Technically, the LP dimension is represented again as a tree on the same set of nodes. Thus, the analytic tree in the PDT format can be built from the ID and LP analyses of the input sentence. The PA dimension could be specified to express (a portion) of the tectogrammatical representation of the sentence.

Only a part of this overall picture has been implemented so far. First, the correspondence between tectogrammatical and analytic levels is quite complicated because some nodes have to be deleted and 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 and LP levels.

2 Description of the Grammar Parts

The experimental XDG grammar induced from PDT utilizes basic principles that are linguistically motivated and traditionally used in many varieties of dependency grammars, including XDG. The current XDG grammar extracted from PDT consists of the following parts: ID Agreement, LP Direction, Simplified ID 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.

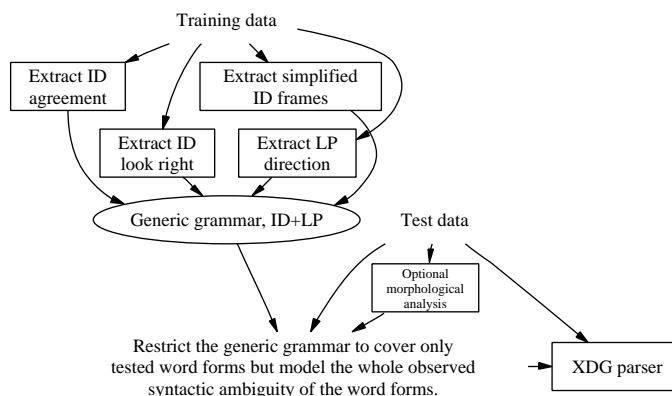


Fig. 2. XDG grammar parts and evaluation.

2.1 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 PDT (1.5 million tokens), there is not enough data to collect syntactic information for individual word forms and even lemmas. Without

a wise technique for combining lexicalized information with some less specific back off, the data sparseness problem is too serious.

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 (such as head-dependent pairs) or n -tuples of tags were examined.

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

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

2.2 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 the 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 the morphology of the father.

This approach ensures grammatical agreement between the father and the daughter and also 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.

2.3 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* [16]. 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. In this first attempt, the LP constraints of the grammar ensure only an acceptable direction (left/right) of an edge between a father and a daughter. The constraints do not model acceptability of different mutual orderings of several daughters of a father.

Technically, the checking of edge direction is implemented by means of topological fields, but these are extremely simplified. Every father at the LP dimension offers three fields: the left and right fields of unlimited cardinality³ and the head field to contain only the father itself. The left field is offered for all the daughters to the left, the head field is used for the father itself and the right field is offered for all the daughters to the right. There is no restriction on mutual ordering of the left or right daughters. The only ensured thing is that every left daughter must precede the father and every right daughter must follow the father.

The LP edge direction is coupled with the ID label of the corresponding ID edge. Given a daughter connected to the father with an ID edge of a particular label, the corresponding LP edge is in certain cases allowed to have only the label LEFT, in other cases only the label RIGHT but sometimes both of the labels (i.e. both directions) are allowed. 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. On the other hand, adverbs can be found both before and after the governing verb and therefore the verb *was* accepts outgoing (ID) edges labelled ADV both in the left and right fields.

An intuitive approach to handle non-projectivities in Czech is to require projective analyses in general but allow for non-projective edges in specific observed cases.⁴ My XDG grammar expresses this requirement in the LP tree only. 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. . . These additional LP labels encode also the ID label, because the syntactic role of the daughter is important with respect to allowing or denying the non-projective realization.

The nodes where a climbed edge may land (such as the word *was* in Figure 3) offer not just the left, head and right fields, but also the required amount of specific X-CLIMBED-Y edges. There is no restriction on mutual linear ordering of the LEFT/RIGHT and *-CLIMBED-* edges.⁵

³ 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.

⁴ Consult [17] for a more advanced approach to restricting non-projectivity.

⁵ This is due to technical limitations of the XDG parser: currently it is not possible to impose partial ordering on topological fields, only linear ordering is supported.

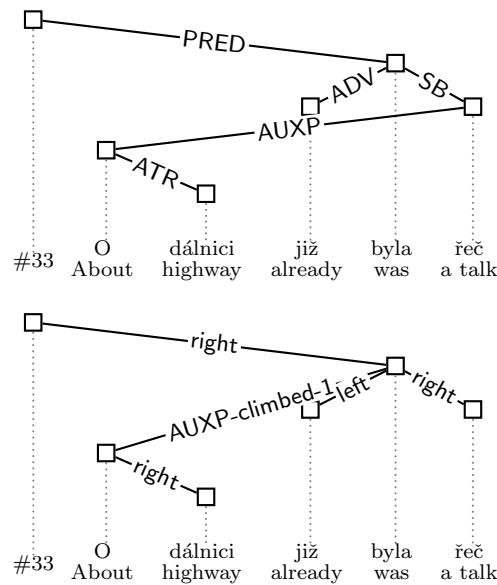


Fig. 3. LP dimension to handle edge direction and non-projectivity.

This way, sentences are analyzed projectively in general, but specific known non-projectivities (based on the simplified morphological tag of the father and the daughter and the ID label of the non-projective edge) are modelled, too.

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.

2.4 Simplified ID 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 Simplified ID 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.

Either the *-CLIMBED-* fields are not mentioned in the ordering at all, or one must specify a full linear ordering among them. It is not possible to express only that all *-CLIMBED-* fields precede the HEAD field without specifying the ordering among them.

Current approaches⁶ aim at distinguishing *complements* vs. *adjuncts*, i.e. modifications that are typically 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, a 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.⁷

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 an adverbial is removed under a verb, the set of modification frames shrinks to a half in size. When the 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.

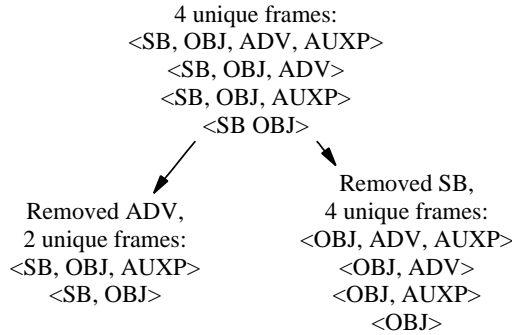
A short example in Figure 5 illustrates the optionality order of modifications observed under infinite verbs (POS=V, SUBPOS=f). In a sample of 2,500 sentences, there were 727 occurrences of infinite verbs. Regardless of the mutual order of modifications of the verbs but with respect to the number of modifications of a particular kind (i.e. representing the modification frame as a multiset, a bag of modification labels), there were 132 unique frames observed. The order of optionality of different modification types is estimated by the described algorithm. The most optional modification (AUXP⁸, a prepositional phrase) is torn off in the first step, reducing the set size from 132 to 95. Equally optional is an adverbial (ADV) because tearing off the adverbial alone would lead to the set size of 95, too. In the cumulative process, the set size after removing AUXP and ADV is 64. The third most optional modification is an object (OBJ) so we

⁶ See [18] for comparison and references.

⁷ 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 all the daughters of a node.

⁸ See [2] for explanation of the labels.

Example: Observed under a verb:



⇒ ADV is more optional than SB.

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

arrive at 46 core frames plus three modifications marked as optional. The resulting frames are shown in Figure 5, too. Each resulting frame contains some fixed members, strictly required at least once, and also the optional modifications with cardinality 0 to the highest observed cardinality of this particular modification. For instance, the first resulting frame $\{AUXP(0-3), ADV(0-3), OBJ(0-2)\}$ contains no fixed members at all, AUXP and ADV are allowed at most 3 times and OBJ is allowed at most twice. Finally, the XDG grammar requires every finite verb in a sentence to satisfy at least one of the allowed modification frames, i.e. to have exactly the allowed number of outgoing ID edges of a particular kind, as the frame prescribes.

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.

2.5 ID Look Right

The generally accepted idea of dependency analysis is that head-daughter dependencies model syntactic analysis best. For German, this assumption is doubted by [19] and it is documented that sister-sister dependencies (lexicalized case) are more informative.

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 view.

Unique observed modification frames: 132
 Set sizes when removing specific modifiers:
 AUXP(95), ADV(95), OBJ(96), AUXC(113), AUXV(119), AUXT(119), AUXX(122),
 COORD(123), SB(126), AUXZ(126), AUXR(126), ATVV(127), PNOM(128),
 AUXG(128), AUXY(129), APOS(129), EXD(130), COORD_PA(131), ATR(131),
 PRED_PA(132), EXD_PA(132)
 Cumulative simplification:
 $132 \rightarrow (\text{AUXP}) \rightarrow 95 \rightarrow (\text{ADV}) \rightarrow 64 \rightarrow (\text{OBJ}) \rightarrow 46$.
 Resulting frames:
 {AUXP(0-3), ADV(0-3), OBJ(0-2)}
 {APOS(1), EXD(1), AUXP(0-3), ADV(0-3), OBJ(0-2)}
 {APOS(1), AUXP(0-3), ADV(0-3), OBJ(0-2)}
 {ATR(2), COORD(1), AUXP(0-3), ADV(0-3), OBJ(0-2)}
 {ATVV(1), AUXC(1), AUXP(0-3), ADV(0-3), OBJ(0-2)}
 {ATVV(1), AUXT(1), AUXP(0-3), ADV(0-3), OBJ(0-2)}
 (... 46 resulting frames altogether)

Fig. 5. Simplifying modifications of infinite verbs.

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

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

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 (as expected for a free word order language), so adding more constraints of this kind is still under consideration.

3 Results

To evaluate the grammar, only the first fixed point in constraint solving is searched. Given a sentence, the 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 more⁹.

⁹ 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

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 PDT and evaluated on 1,800 to 2,000 unseen sentences from the standard evaluation part of PDT (devtest). The results are displayed in Table 4.

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.

Note that the number of training sentences was relatively low (around 2 to 5% of PDT), which explains the relatively high number of unsolved sentences (around 10 to 20%). A wider coverage of the grammar is achieved by training on more data but immediately leads to significant growth of the number of solutions available. This problem with scalability can be solved only by providing the grammar with more constraints of various kinds. 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

variables and starts propagating constraints again. Other 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.

edges, around 85% is correct. Based on other experiments, training on more data leads to a lower error rate but less edges securely established.

Contrary to our expectations, 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. One possible explanation comes from the word order freedom of Czech. The Look Right principle probably helps to establish rigid structures such as dates but leads to wrong decisions in general, because it faces a serious sparse data problem. A deeper analysis is necessary to confirm this explanation.

4 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 current grammar relies on very few types of constraints. More constraints of different kinds have to be added to achieve both a better scalability of the grammar and a more effective propagation of the constraints.¹⁰ The current grammar lacks especially such a kind of constraints that bind information together—the current constraints are too independent to achieve strong 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.

Second, there are several kinds of expressions that in fact have no dependency structure, such as names, dates and other multi-word expressions. Coordination should be handled specifically, too. The “dependency” analysis of such expressions in 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. This expectation comes from my analysis of sources of structural ambiguity modelled by the grammar: given the set of all trees assigned by the grammar to a string of words, punctuation symbols, cardinals, adverbs and conjunctions (in this order) are the parts of speech that have most different governors.

Third, the tested version of XDG parser could not make any use of frequency information contained in PDT.¹¹ The possibility to guide the XDG parser by frequency information to find a plausible solution sooner is explored in [21] but the research is still in progress.¹²

¹⁰ Similarly as [20] observed for English, purely syntactic constraints are too weak to analyse Czech. The deep syntactic level of PDT and the Czech valency lexicon provide a promising source of additional constraints.

¹¹ In an experiment, frequency information was used as a threshold to ignore rare edge assignments. The thresholding resulted in lower coverage *and* lower precision.

¹² A similar constraint-based dependency parsing by [22] inherently includes weight of constraints, but no directly comparable results were published so far. [23] report

5 Conclusion

I described an experiment with constraint based dependency parsing of a language with rich morphology and relatively free 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.

6 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. This work has been partially supported by the Ministry of Education of the Czech Republic, project LN00A063.

References

1. Sgall, P., Hajičová, E., Panevová, J.: The Meaning of the Sentence and Its Semantic and Pragmatic Aspects. Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands (1986)
2. Hajič, J., Panevová, J., Buráňová, E., Urešová, Z., Bémová, A.: A Manual for Analytic Layer Tagging of the Prague Dependency Treebank. Technical Report TR-2001-, ÚFAL MFF UK, Prague, Czech Republic (2001) English translation of the original Czech version.
3. Hajičová, E., Panevová, J., Sgall, P.: A Manual for Tectogrammatic Tagging of the Prague Dependency Treebank. Technical Report TR-2000-09, ÚFAL MFF UK, Prague, Czech Republic (2000) In Czech.
4. Žabokrtský, Z., Benešová, V., Lopatková, M., Skwarská, K.: Tektogramaticky anotovaný valenční slovník českých sloves. Technical Report TR-2002-15, ÚFAL/CKL, Prague, Czech Republic (2002)
5. Collins, M., Hajič, J., Brill, E., Ramshaw, L., Tillmann, C.: A Statistical Parser of Czech. In: Proceedings of 37th ACL Conference, University of Maryland, College Park, USA (1999) 505–512
6. Zeman, D.: Can Subcategorization Help a Statistical Parser? In: Proceedings of the 19th International Conference on Computational Linguistics (Coling 2002), Taipei, Tchaj-wan, Zhongyang Yanjiuyuan (Academia Sinica) (2002)
7. Charniak, E.: A Maximum-Entropy-Inspired Parser. In: Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-2000), Seattle, Washington, USA (2000) 132–139 <http://www.cs.brown.edu/people/ec/papers/>.

edge accuracy of 96.63% on a corpus of 200 sentences with average length 8.8 words, significantly less than in our data.

8. Debusmann, R., Duchier, D., Koller, A., Kuhlmann, M., Smolka, G., Thater, S.: A Relational Syntax-Semantics Interface Based on Dependency Grammar. In: Proceedings of COLING 2004, Geneva, Switzerland (2004)
9. Duchier, D., Debusmann, R.: Topological Dependency Trees: A Constraint-based Account of Linear Precedence. In: 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001). (2001)
10. Debusmann, R., Duchier, D.: A Meta-Grammatical Framework for Dependency Grammar (2003)
11. Bojar, O.: Czech Syntactic Analysis Constraint-Based, XDG: One Possible Start. Prague Bulletin of Mathematical Linguistics (2004) 43–54
12. Holan, T.: K syntaktické analýze českých(!) vět. In: MIS 2003, MATFYZPRESS (2003)
13. Yamada, H., Matsumoto, Y.: Statistical dependency analysis with support vector machines. In: Proceedings of the International Workshop on Parsing Technologies (IWPT 2003), Nancy, France (2003)
14. Bojar, O.: Towards Automatic Extraction of Verb Frames. Prague Bulletin of Mathematical Linguistics (2003) 101–120
15. Kruijff, G.J.M.: 3-Phase Grammar Learning. In: Proceedings of the Workshop on Ideas and Strategies for Multilingual Grammar Development. (2003)
16. Bech, G.: Studien über das deutsche Verbum infinitum. (1955) 2nd unrevised edition published 1983 by Max Niemeyer Verlag, Tübingen (Linguistische Arbeiten 139).
17. Holan, T., Kuboň, V., Oliva, K., Plátek, M.: Two Useful Measures of Word Order Complexity. In Polguere, A., Kahane, S., eds.: Proceedings of the Coling '98 Workshop: Processing of Dependency-Based Grammars, Montreal, University of Montreal (1998)
18. Sarkar, A., Zeman, D.: Automatic Extraction of Subcategorization Frames for Czech. In: Proceedings of the 18th International Conference on Computational Linguistics (Coling 2000), Saarbrücken, Germany, Universität des Saarlandes (2000)
19. Dubey, A., Keller, F.: Probabilistic Parsing for German using Sister-Head Dependencies. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo (2003) 96–103
20. Harper, M.P., Helzerman, R.A., Zoltowski, C.B., Yeo, B.L., Chan, Y., Stewart, T., Pellom, B.L.: Implementation Issues in the Development of the PARSEC Parser. SOFTWARE - Practice and Experience **25** (1995) 831–862
21. Dienes, P., Koller, A., Kuhlmann, M.: Statistical A-Star Dependency Parsing. In Duchier, D., ed.: Prospects and Advances of the Syntax/Semantics Interface, Nancy (2003) 85–89
22. Heinecke, J., Kunze, J., Menzel, W., Schöder, I.: Eliminative parsing with graded constraints. In: Proceedings of COLING-ACL Conference, Montreal, Canada (1998)
23. Foth, K., Menzel, W., Schröder, I.: Robust Parsing with Weighted Constraints. Natural Language Engineering (2004) in press.