Beyond Projectivity: Multilingual Evaluation of Constraints and Measures on Non-Projective Structures

Jiří Havelka
Institute of Formal and Applied Linguistics
Charles University in Prague
Czech Republic
havelka@ufal.mff.cuni.cz

Abstract

Dependency analysis of natural language has gained importance for its applicability to NLP tasks. Non-projective structures are common in dependency analysis, therefore we need fine-grained means of describing them, especially for the purposes of machine-learning oriented approaches like parsing. We present an evaluation on twelve languages which explores several constraints and measures on non-projective structures. We pursue an edge-based approach concentrating on properties of individual edges as opposed to properties of whole trees. In our evaluation, we include previously unreported measures taking into account levels of nodes in dependency trees. Our empirical results corroborate theoretical results and show that an edge-based approach using levels of nodes provides an accurate and at the same time expressive means for capturing non-projective structures in natural language.

1 Introduction

Dependency analysis of natural language has been gaining an ever increasing interest thanks to its applicability in many tasks of NLP—a recent example is the dependency parsing work of McDonald et al. (2005), which introduces an approach based on the search for maximum spanning trees, capable of handling non-projective structures naturally.

The study of dependency structures occurring in natural language can be approached from two sides: by trying to delimit permissible dependency structures through formal constraints (for a recent review paper, see Kuhlmann and Nivre (2006)), or by providing their linguistic description (see e.g. Veselá et al. (2004) and Hajičová et al. (2004) for a linguistic analysis of non-projective constructions in Czech.[1])

We think that it is worth bearing in mind that neither syntactic structures in dependency treebanks, nor structures arising in machine-learning approaches, such as MST dependency parsing, need a priori fall into any formal subclass of dependency trees. We should therefore aim at formal means capable of describing all non-projective structures that are both expressive and fine-grained enough to be useful in statistical approaches, and at the same time suitable for an adequate linguistic description.[2]

Holan et al. (1998) first defined an infinite hierarchy of classes of dependency trees, going from projective to unrestricted dependency trees, based on the notion of gap degree for subtrees (cf. Section 3). Holan et al. (2000) present linguistic considerations concerning Czech and English with respect to this hierarchy (cf. also Section 6).

In this paper, we consider all constraints and measures evaluated by Kuhlmann and Nivre (2006)—with some minor variations, cf. Section 4.2. Ad-

[1] These two papers contain an error concerning an alternative condition of projectivity, which is rectified in Havelka (2005).

[2] The importance of such means becomes more evident from the asymptotically negligible proportion of projective trees to all dependency trees; there are super-exponentially many unrestricted trees compared to exponentially many projective trees on n nodes. Unrestricted dependency trees (i.e. labelled rooted trees) and projective dependency trees are counted by sequences A000169 and A006013 (offset 1), respectively, in the On-Line Encyclopedia of Sequences (Sloane, 2007).

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 608–615,
Prague, Czech Republic, June 2007. ©2007 Association for Computational Linguistics

ditionally, we introduce several measures not considered in their work. We also extend the empirical basis from Czech and Danish to twelve languages, which were made available in the CoNLL-X shared task on dependency parsing.

In our evaluation, we do not address the issue of what possible effects the annotations and/or conversions used when creating the data might have on non-projective structures in the different languages.

The newly considered measures have the first or both of the following desiderata: they are based on properties of individual non-projective edges (cf. Definition 3); and they take into account levels of nodes in dependency trees explicitly. None of the constraints and measures in Kuhlmann and Nivre (2006) take into account levels of nodes explicitly.

Level types of non-projective edges, introduced by Havelka (2005), have both desiderata. They provide an edge-based means of characterizing all non-projective structures; they also have some further interesting formal properties.

We propose a novel, more detailed measure, *level signatures* of non-projective edges, combining levels of nodes with the partitioning of gaps of non-projective edges into components. We derive a formal property of these signatures that links them to the constraint of well-nestedness, which is an extension of the result for level types (see also Havelka (2007b)).

The paper is organized as follows: Section 2 contains formal preliminaries; in Section 3 we review the constraint of projectivity and define related notions necessary in Section 4, where we define and discuss all evaluated constraints and measures; Section 5 describes our data and experimental setup; empirical results are presented in Section 6.

2 Formal preliminaries

Here we provide basic definitions and notation used in subsequent sections.

Definition 1 A *dependency tree* is a triple $(V, \rightarrow, \preceq)$, where V is a finite set of nodes, $\rightarrow$ a *dependency* relation on V, and $\preceq$ a total order on V.[3]

[3] We adopt the following convention: nodes are drawn top-down according to their increasing level, with nodes on the same level being the same distance from the root; nodes are drawn from left to right according to the total order on nodes; edges are drawn as solid lines, paths as dotted curves.

Relation $\rightarrow$ models linguistic dependency, and so represents a directed, rooted tree on V. There are many ways of characterizing rooted trees, we give here a characterization via the properties of $\rightarrow$: there is a *root* $r \in V$ such that $r \rightarrow^* v$ for all $v \in V$ and there is a unique *edge* $p \rightarrow v$ for all $v \in V$, $v \neq r$, and no edge into r. Relation $\rightarrow^*$ is the reflexive transitive closure of $\rightarrow$ and is usually called *subordination*.

For each node i we define its *level* as the length of the path $r \rightarrow^* i$; we denote it level_i. The symmetrization $\leftrightarrow = \rightarrow \cup \rightarrow^{-1}$ makes it possible to talk about edges (pairs of nodes i, j such that $i \rightarrow j$) without explicitly specifying the *parent* (*head*; i here) and the *child* (*dependent*; j here); so $\rightarrow$ represents directed edges and $\leftrightarrow$ undirected edges. To retain the ability to talk about the direction of edges, we define $\mathsf{Parent}_{i \leftrightarrow j} = \begin{cases} i & \text{if } i \rightarrow j \\ j & \text{if } j \rightarrow i \end{cases}$ and $\mathsf{Child}_{i \leftrightarrow j} = \begin{cases} j & \text{if } i \rightarrow j \\ i & \text{if } j \rightarrow i \end{cases}$. To make the exposition clearer by avoiding overuse of the symbol $\rightarrow$, we introduce notation for rooted *subtrees* not only for nodes, but also for edges: $\mathsf{Subtree}_i = \{v \in V \mid i \rightarrow^* v\}$, $\mathsf{Subtree}_{i \leftrightarrow j} = \{v \in V \mid \mathsf{Parent}_{i \leftrightarrow j} \rightarrow^* v\}$ (note that the subtree of an edge is defined relative to its parent node). To be able to talk concisely about the total order on nodes $\preceq$, we define *open intervals* whose endpoints need not be in a prescribed order $(i, j) = \{v \in V \mid \min_{\preceq}\{i, j\} \prec v \prec \max_{\preceq}\{i, j\}\}$.

3 Condition of projectivity

Projectivity of a dependency tree can be characterized both through the properties of its subtrees and through the properties of its edges.[4]

Definition 2 A dependency tree $T = (V, \rightarrow, \preceq)$ is *projective* if it satisfies the following equivalent conditions:

(Harper & Hays)
$i \rightarrow j \ \& \ v \in (i, j) \Longrightarrow v \in \mathsf{Subtree}_i$,

(Lecerf & Ihm)
$j \in \mathsf{Subtree}_i \ \& \ v \in (i, j) \Longrightarrow v \in \mathsf{Subtree}_i$,

(Fitialov)
$j_1, j_2 \in \mathsf{Subtree}_i \ \& \ v \in (j_1, j_2) \Longrightarrow v \in \mathsf{Subtree}_i$.

Otherwise T is *non-projective*.

[4] There are many other equivalent characterizations of projectivity, we give only three historically prominent ones.

It was Marcus (1965) who proved the equivalence of the conditions in Definition 2, proposed in the early 1960's (we denote them by the names of those to whom Marcus attributes their authorship).

We see that the antecedents of the projectivity conditions move from edge-focused to subtree-focused (i.e. from talking about dependency to talking about subordination).

It is the condition of Fitialov that has been mostly explored when studying so-called relaxations of projectivity. (The condition is usually worded as follows: A dependency tree is projective if the nodes of all its subtrees constitute contiguous intervals in the total order on nodes.)

However, we find the condition of Harper & Hays to be the most appealing from the linguistic point of view because it gives prominence to the primary notion of dependency edges over the derived notion of subordination. We therefore use an edge-based approach whenever we find it suitable.

To that end, we need the notion of a non-projective edge and its gap.

Definition 3 For any edge $i \leftrightarrow j$ in a dependency tree T we define its *gap* as follows

$$\mathsf{Gap}_{i\leftrightarrow j} = \{v \in V \mid v \in (i,j) \ \& \ v \notin \mathsf{Subtree}_{i\leftrightarrow j}\} \ .$$

An edge with an empty gap is *projective*, an edge whose gap is non-empty is *non-projective*.[5]

We see that non-projective are those edges $i \leftrightarrow j$ for which there is a node v such that together they violate the condition of Harper & Hays; we group all such nodes v into $\mathsf{Gap}_{i\leftrightarrow j}$, the gap of the non-projective edge $i \leftrightarrow j$.

The notion of gap is defined differently for subtrees of a dependency tree (Holan et al., 1998; Bodirsky et al., 2005). There it is defined through the nodes of the whole dependency tree not in the considered subtree that intervene between its nodes in the total order on nodes $\preceq$.

4 Relaxations of projectivity: evaluated constraints and measures

In this section we present all constraints and measures on dependency trees that we evaluate empirically in Section 6. First we give definitions of global constraints on dependency trees, then we present measures of non-projectivity based on properties of individual non-projective edges (some of the edge-based measures have corresponding tree-based counterparts, however we do not discuss them in detail).

4.1 Tree constraints

We consider the following three global constraints on dependency trees: projectivity, planarity, and well-nestedness. All three constraints can be applied to more general structures, e.g. dependency forests or even general directed graphs. Here we adhere to their primary application to dependency trees.

Definition 4 A dependency tree T is *non-planar* if there are two edges $i_1 \leftrightarrow j_1$, $i_2 \leftrightarrow j_2$ in T such that

$$i_1 \in (i_2, j_2) \ \& \ i_2 \in (i_1, j_1) \ .$$

Otherwise T is *planar*.

Planarity is a relaxation of projectivity that corresponds to the "no crossing edges" constraint. Although it might get confused with projectivity, it is in fact a strictly weaker constraint. Planarity is equivalent to projectivity for dependency trees with their root node at either the left or right fringe of the tree.

Planarity is a recent name for a constraint studied under different names already in the 1960's—we are aware of independent work in the USSR (*weakly non-projective trees*; see the survey paper by Dikovsky and Modina (2000) for references) and in Czechoslovakia (*smooth trees*; Nebeský (1979) presents a survey of his results).

Definition 5 A dependency tree T is *ill-nested* if there are two non-projective edges $i_1 \leftrightarrow j_1$, $i_2 \leftrightarrow j_2$ in T such that

$$i_1 \in \mathsf{Gap}_{i_2\leftrightarrow j_2} \ \& \ i_2 \in \mathsf{Gap}_{i_1\leftrightarrow j_1} \ .$$

Otherwise T is *well-nested*.

Well-nestedness was proposed by Bodirsky et al. (2005). The original formulation forbids interleaving of disjoint subtrees in the total order on nodes; we present an equivalent formulation in terms of non-projective edges, derived in (Havelka, 2007b).

Figure 1 illustrates the subset hierarchy between classes of dependency trees satisfying the particular constraints:

$$\text{projective} \subsetneq \text{planar} \subsetneq \text{well-nested} \subsetneq \text{unrestricted}$$

[5] In figures with sample configurations we adopt this convention: for a non-projective edge, we draw all nodes in its gap explicitly and assume that no node on any path crossing the span of the edge lies in the interval delimited by its endpoints.

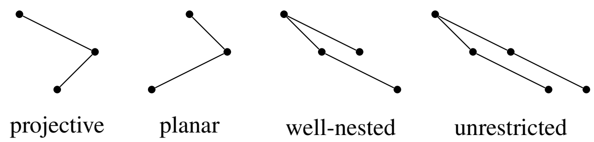

Figure 1: Sample dependency trees (trees satisfy corresponding constraints and violate all preceding ones)

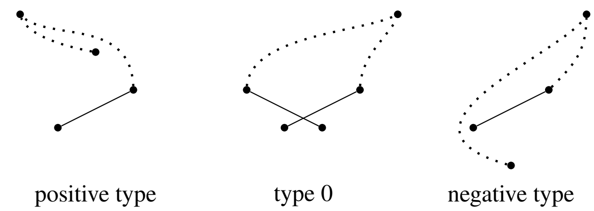

Figure 2: Sample configurations with non-projective edges of different level types

4.2 Edge measures

The first two measures are based on two ways of partitioning the gap of a non-projective edge—into intervals and into components. The third measure, level type, is based on levels of nodes. We also propose a novel measure combining levels of nodes and the partitioning of gaps into components.

Definition 6 For any edge $i \leftrightarrow j$ in a dependency tree T we define its *interval degree* as follows

$$\mathsf{ideg}_{i\leftrightarrow j} = \text{number of intervals in } \mathsf{Gap}_{i\leftrightarrow j} \ .$$

By an interval we mean a contiguous interval in $\preceq$, i.e. a maximal set of nodes comprising all nodes between its endpoints in the total order on nodes $\preceq$.

This measure corresponds to the tree-based *gap degree* measure in (Kuhlmann and Nivre, 2006), which was first introduced in (Holan et al., 1998)—there it is defined as the maximum over gap degrees of all subtrees of a dependency tree (the gap degree of a subtree is the number of contiguous intervals in the gap of the subtree). The interval degree of an edge is bounded from above by the gap degree of the subtree rooted in its parent node.

Definition 7 For any edge $i \leftrightarrow j$ in a dependency tree T we define its *component degree* as follows

$$\mathsf{cdeg}_{i\leftrightarrow j} = \text{number of components in } \mathsf{Gap}_{i\leftrightarrow j} \ .$$

By a component we mean a connected component in the relation $\leftrightarrow$, in other words a weak component in the relation $\rightarrow$ (we consider relations induced on the set $\mathsf{Gap}_{i\leftrightarrow j}$ by relations on T).

This measure was introduced by Nivre (2006); Kuhlmann and Nivre (2006) call it *edge degree*. Again, they define it as the maximum over all edges.

Each component of a gap can be represented by a single node, its *root* in the dependency relation induced on the nodes of the gap (i.e. a node of the component closest to the root of the whole tree). Note that a component need not constitute a full subtree of the dependency tree (there may be nodes in the subtree of the component root that lie outside the span of the particular non-projective edge).

Definition 8 The *level type* (or just *type*) of a non-projective edge $i \leftrightarrow j$ in a dependency tree T is defined as follows

$$\mathsf{Type}_{i\leftrightarrow j} = \mathsf{level}_{\mathsf{Child}_{i\leftrightarrow j}} - \min_{n\in\mathsf{Gap}_{i\leftrightarrow j}} \mathsf{level}_n \ .$$

The level type of an edge is the relative distance in levels of its child node and a node in its gap closest to the root; there may be more than one node witnessing an edge's type. For sample configurations see Figure 2. Properties of level types are presented in Havelka (2005; 2007b).[6]

We propose a new measure combining level types and component degrees. (We do not use interval degrees, i.e. the partitioning of gaps into intervals, because we cannot specify a unique representative of an interval with respect to the tree structure.)

Definition 9 The *level signature* (or just *signature*) of an edge $i \leftrightarrow j$ in a dependency tree T is a mapping $\mathsf{Signature}_{i\leftrightarrow j} : \mathcal{P}(V) \rightarrow {}^{\mathbb{Z}}\mathbb{N}_0$ defined as follows

$$\mathsf{Signature}_{i\leftrightarrow j} = \{\mathsf{level}_{\mathsf{Child}_{i\leftrightarrow j}} - \mathsf{level}_r \mid r \text{ is component root in } \mathsf{Gap}_{i\leftrightarrow j}\} \ .$$

(The right-hand side is considered as a multiset, i.e. elements may repeat.) We call the elements of a signature *component levels*.

The signature of an edge is a multiset consisting of the relative distances in levels of all component roots in its gap from its child node.

Further, we disregard any possible orderings on signatures and concentrate only on the relative distances in levels. We present signatures as non-

[6] For example, presence of non-projective edges of nonnegative level type in equivalent to non-projectivity of a dependency tree; moreover, all such edges can be found in linear time.

decreasing sequences and write them in angle brackets $\langle\ \rangle$, component levels separated by commas (by doing so, we avoid combinatorial explosion).

Notice that level signatures subsume level types: the level type of a non-projective edge is the component level of any of possibly several component roots closest to the root of the whole tree. In other words, the level type of an edge is equal to the largest component level occurring in its level signature.

Level signatures share interesting formal properties with level types of non-projective edges. The following result is a direct extension of the results presented in Havelka (2005; 2007b).

Theorem 10 *Let $i \leftrightarrow j$ be a non-projective edge in a dependency tree T. For any component c in $\mathsf{Gap}_{i\leftrightarrow j}$ represented by root r_c with component level $l_c \leq 0$ (< 0) there is a non-projective edge $v \to r_c$ in T with $\mathsf{Type}_{v\leftrightarrow r_c} \geq 0$ (> 0) such that either $i \in \mathsf{Gap}_{v\leftrightarrow r_c}$, or $j \in \mathsf{Gap}_{v\leftrightarrow r_c}$.*

PROOF. From the assumptions $l_c \leq 0$ and $r_c \in \mathsf{Gap}_{i\leftrightarrow j}$ the parent v of node r_c lies outside the span of the edge $i \leftrightarrow j$, hence $v \notin \mathsf{Gap}_{i\leftrightarrow j}$. Thus either $i \in (v, r_c)$, or $j \in (v, r_c)$. Since $\mathsf{level}_v \geq \mathsf{level}_{\mathsf{Parent}_{i\leftrightarrow j}}$, we have that $\mathsf{Parent}_{i\leftrightarrow j} \notin \mathsf{Subtree}_v$, and so either $i \in \mathsf{Gap}_{v\leftrightarrow r_c}$, or $j \in \mathsf{Gap}_{v\leftrightarrow r_c}$. Finally from $l_c = \mathsf{level}_{\mathsf{Child}_{i\leftrightarrow j}} - \mathsf{level}_{r_c} \leq 0$ (< 0) we get $\mathsf{level}_{r_c} - \mathsf{level}_{\mathsf{Child}_{i\leftrightarrow j}} \geq 0$ (> 0), hence $\mathsf{Type}_{v\leftrightarrow r_c} \geq 0$ (> 0). ∎

This result links level signatures to well-nestedness: it tells us that whenever an edge's signature contains a nonpositive component level, the whole dependency tree is ill-nested (because then there are two edges satisfying Definition 5).

All discussed edge measures take integer values: interval and component degrees take only nonnegative values, level types and level signatures take integer values (in all cases, their absolute values are bounded by the size of the whole dependency tree). Both interval and component degrees are defined also for projective edges (for which they take value 0), level type is undefined for projective edges, however the level signature of projective edges is defined—it is the empty multiset/sequence.

5 Data and experimental setup

We evaluate all constraints and measures described in the previous section on 12 languages, whose treebanks were made available in the CoNLL-X shared task on dependency parsing (Buchholz and Marsi, 2006). In alphabetical order they are: Arabic, Bulgarian, Czech, Danish, Dutch, German, Japanese, Portuguese, Slovene, Spanish, Swedish, and Turkish (Hajič et al., 2004; Simov et al., 2005; Böhmová et al., 2003; Kromann, 2003; van der Beek et al., 2002; Brants et al., 2002; Kawata and Bartels, 2000; Afonso et al., 2002; Džeroski et al., 2006; Civit Torruella and Martí Antonín, 2002; Nilsson et al., 2005; Oflazer et al., 2003).[7] We do not include Chinese, which is also available in this data format, because all trees in this data set are projective.

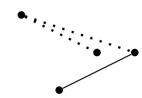

Figure 3: Sample non-projective tree considered planar in empirical evaluation

We take the data "as is", although we are aware that structures occurring in different languages depend on the annotations and/or conversions used (some languages were not originally annotated with dependency syntax, but only converted to a unified dependency format from other representations).

The CoNLL data format is a simple tabular format for capturing dependency analyses of natural language sentences. For each sentence, it uses a technical root node to which dependency analyses of parts of the sentence (possibly several) are attached. Equivalently, the representation of a sentence can be viewed as a forest consisting of dependency trees.

By conjoining partial dependency analyses under one technical root node, we let all their edges interact. Since the technical root comes before the sentence itself, no new non-projective edges are introduced. However, edges from technical roots may introduce non-planarity. Therefore, in our empirical evaluation we disregard all such edges when counting trees conforming to the planarity constraint; we also exclude them from the total numbers of edges. Figure 3 exemplifies how this may affect counts of non-planar trees;[8] cf. also the remark after Definition 4. Counts of well-nested trees are not affected.

[7] All data sets are the train parts of the CoNLL-X shared task.

[8] The sample tree is non-planar according to Definition 4, however we do not consider it as such, because all pairs of "crossing edges" involve an edge from the technical root (edges from the technical root are depicted as dotted lines).

6 Empirical results

Our complete results for global constraints on dependency trees are given in Table 1. They confirm the findings of Kuhlmann and Nivre (2006): planarity seems to be almost as restrictive as projectivity; well-nestedness, on the other hand, covers large proportions of trees in all languages.

In contrast to global constraints, properties of individual non-projective edges allow us to pinpoint the causes of non-projectivity. Therefore they provide a means for a much more fine-grained classification of non-projective structures occurring in natural language. Table 2 presents highlights of our analysis of edge measures.

Both interval and component degrees take generally low values. On the other hand, Holan et al. (1998; 2000) show that at least for Czech neither of these two measures can in principle be bounded.

Taking levels of nodes into account seems to bring both better accuracy and expressivity. Since level signatures subsume level types as their last components, we only provide counts of edges of positive, nonpositive, and negative level types. For lack of space, we do not present full distributions of level types nor of level signatures.

Positive level types give an even better fit with real linguistic data than the global constraint of well-nestedness (an ill-nested tree need not contain a non-projective edge of nonpositive level type; cf. Theorem 10). For example, in German less than one tenth of ill-nested trees contain an edge of nonpositive level type. Minimum negative level types for Czech, Slovene, Swedish, and Turkish are respectively -1, -5, -2, and -4.

Level signatures combine level types and component degrees, and so give an even more detailed picture of the gaps of non-projective edges. In some languages the actually occurring signatures are quite limited, in others there is a large variation.

Because we consider it linguistically relevant, we also count how many non-projective edges contain in their gaps a component rooted in an ancestor of the edge (an *ancestor* of an edge is any node on the path from the root of the whole tree to the parent node of the edge). The proportions of such non-projective edges vary widely among languages and for some this property seems highly important.

Empirical evidence shows that edge measures of non-projectivity taking into account levels of nodes fit very well with linguistic data. This supports our theoretical results and confirms that properties of non-projective edges provide a more accurate as well as expressive means for describing non-projective structures in natural language than the constraints and measures considered by Kuhlmann and Nivre (2006).

7 Conclusion

In this paper, we evaluate several constraints and measures on non-projective dependency structures. We pursue an edge-based approach giving prominence to properties of individual edges. At the same time, we consider levels of nodes in dependency trees. We find an edge-based approach also more appealing linguistically than traditional approaches based on properties of whole dependency trees or their subtrees. Furthermore, edge-based properties allow machine-learning techniques to model global phenomena locally, resulting in less sparse models.

We propose a new edge measure of non-projectivity, level signatures of non-projective edges. We prove that, analogously to level types, they relate to the constraint of well-nestedness.

Our empirical results on twelve languages can be summarized as follows: Among the global constraints, well-nestedness fits best with linguistic data. Among edge measures, the previously unreported measures taking into account levels of nodes stand out. They provide both the best fit with linguistic data of all constraints and measures we have considered, as well as a substantially more detailed capability of describing non-projective structures.

The interested reader can find a more in-depth and broader-coverage discussion of properties of dependency trees and their application to natural language syntax in (Havelka, 2007a).

As future work, we plan to investigate more languages and carry out linguistic analyses of non-projective structures in some of them. We will also apply our results to statistical approaches to NLP tasks, such as dependency parsing.

Acknowledgement The research reported in this paper was supported by Project No. 1ET201120505 of the Ministry of Education of the Czech Republic.

Language	Arabic	Bulgarian	Czech	Danish	Dutch	German	Japanese	Portuguese	Slovene	Spanish	Swedish	Turkish
ill-nested	1		79	6	15	416		7	3		71	14
non-planar	150	677	13783	787	4115	10865	1	1713	283	56	1076	556
non-projective	163	690	16831	811	4865	10883	902	1718	340	57	1079	580
proportion of all (%)	11.16%	5.38%	23.15%	15.63%	36.44%	27.75%	5.29%	18.94%	22.16%	1.72%	9.77%	11.6%
all	1460	12823	72703	5190	13349	39216	17044	9071	1534	3306	11042	4997

Table 1: Counts of dependency trees violating global constraints of well-nestedness, planarity, and projectivity; the last line gives the total numbers of dependency trees. (An empty cell means count zero.)

Language	Arabic	Bulgarian	Czech	Danish	Dutch	German	Japanese	Portuguese	Slovene	Spanish	Swedish	Turkish
ideg $= 1$	211	724	23376	940	10209	14605	1570	2398	548	58	1829	813
ideg $= 2$		1	189	5	349	1198	81	272	2	1	46	27
ideg $= 3$			3		8	37	12	24			9	1
cdeg $= 1$	200	723	23190	842	10264	13107	1484	2466	531	59	1546	623
cdeg $= 2$	10	1	292	78	238	2206	143	151	11		204	146
cdeg $= 3$	1	1	66	22	47	434	26	64	2		76	55
Type > 0	211	725	23495	942	10564	15803	1667	2699	547	59	1847	833
Type ≤ 0			75	3	2	41		3	3		50	8
Type < 0			4						2		15	2
Signature / count	$\langle 1\rangle$ / 92	$\langle 2\rangle$ / 674	$\langle 2\rangle$ / 18507	$\langle 2\rangle$ / 555	$\langle 2\rangle$ / 8061	$\langle 2\rangle$ / 8407	$\langle 1\rangle$ / 466	$\langle 2\rangle$ / 1670	$\langle 2\rangle$ / 384	$\langle 2\rangle$ / 46	$\langle 2\rangle$ / 823	$\langle 2\rangle$ / 341
	$\langle 2\rangle$ / 56	$\langle 3\rangle$ / 32	$\langle 1\rangle$ / 2886	$\langle 1\rangle$ / 115	$\langle 3\rangle$ / 1461	$\langle 1\rangle$ / 3112	$\langle 2\rangle$ / 209	$\langle 1\rangle$ / 571	$\langle 1\rangle$ / 67	$\langle 3\rangle$ / 7	$\langle 1\rangle$ / 530	$\langle 1\rangle$ / 189
	$\langle 3\rangle$ / 18	$\langle 1\rangle$ / 10	$\langle 3\rangle$ / 1515	$\langle 3\rangle$ / 100	$\langle 1\rangle$ / 512	$\langle 1,1\rangle$ / 1503	$\langle 4\rangle$ / 186	$\langle 3\rangle$ / 208	$\langle 3\rangle$ / 45	$\langle 4\rangle$ / 4	$\langle 3\rangle$ / 114	$\langle 1,1\rangle$ / 91
	$\langle 4\rangle$ / 10	$\langle 4\rangle$ / 5	$\langle 4\rangle$ / 154	$\langle 1,1\rangle$ / 63	$\langle 4\rangle$ / 201	$\langle 3\rangle$ / 1397	$\langle 3\rangle$ / 183	$\langle 1,1\rangle$ / 113	$\langle 4\rangle$ / 13	$\langle 1\rangle$ / 2	$\langle 1,1\rangle$ / 94	$\langle 3\rangle$ / 53
	$\langle 1,1\rangle$ / 8	$\langle 5\rangle$ / 2	$\langle 1,1\rangle$ / 115	$\langle 4\rangle$ / 41	$\langle 1,1\rangle$ / 118	$\langle 2,2\rangle$ / 476	$\langle 5\rangle$ / 126	$\langle 1,1,1\rangle$ / 44	$\langle 5\rangle$ / 12		$\langle 0\rangle$ / 31	$\langle 2,2\rangle$ / 31
	$\langle 5\rangle$ / 7	$\langle 1,1,1\rangle$ / 1	$\langle 0\rangle$ / 70	$\langle 5\rangle$ / 16	$\langle 2,2\rangle$ / 52	$\langle 1,1,1\rangle$ / 312	$\langle 6\rangle$ / 113	$\langle 2,2\rangle$ / 29	$\langle 1,1\rangle$ / 6		$\langle 1,3\rangle$ / 27	$\langle 1,1,1\rangle$ / 29
	$\langle 6\rangle$ / 6	$\langle 1,1\rangle$ / 1	$\langle 2,2\rangle$ / 58	$\langle 1,1,1\rangle$ / 16	$\langle 1,1,1\rangle$ / 25	$\langle 4\rangle$ / 136	$\langle 7\rangle$ / 78	$\langle 2,2,2\rangle$ / 13	$\langle 6\rangle$ / 4		$\langle 1,1,1\rangle$ / 25	$\langle 4\rangle$ / 19
	$\langle 7\rangle$ / 4		$\langle 1,1,1\rangle$ / 48	$\langle 2,2\rangle$ / 7	$\langle 5\rangle$ / 23	$\langle 3,3\rangle$ / 98	$\langle 1,1\rangle$ / 63	$\langle 4\rangle$ / 12	$\langle 1,1,1,1\rangle$ / 4		$\langle 4\rangle$ / 21	$\langle 2,2,2\rangle$ / 10
	$\langle 2,2\rangle$ / 2		$\langle 2,4\rangle$ / 44	$\langle 6\rangle$ / 6	$\langle 1,3\rangle$ / 16	$\langle 2,2,2\rangle$ / 69	$\langle 8\rangle$ / 49	$\langle 1,1,1,1\rangle$ / 7	$\langle 7\rangle$ / 2		$\langle 1,2\rangle$ / 19	$\langle 3,3\rangle$ / 6
	$\langle 9\rangle$ / 1		$\langle 1,3\rangle$ / 32	$\langle 2,2,2\rangle$ / 6	$\langle 3,3\rangle$ / 15	$\langle 1,1,1,1\rangle$ / 59	$\langle 9\rangle$ / 35	$\langle 1,1,1,1,1\rangle$ / 6	$\langle 1,1,3\rangle$ / 2		$\langle 2,2\rangle$ / 16	$\langle 2,2,2,2\rangle$ / 6
	⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮	⋮
ancestor comp. root	39	711	20035	703	9781	10128	0	1832	392	57	950	345
only ancestor comp. r.	39	711	19913	685	9697	9526	0	1820	386	57	857	340
non-projective	211	725	23570	945	10566	15844	1667	2702	550	59	1897	841
proportion of all (%)	0.42%	0.41%	2.13%	1.06%	5.9%	2.4%	1.32%	1.37%	2.13%	0.07%	1.05%	1.61%
all	50097	177394	1105437	89171	179063	660394	126511	197607	25777	86028	180425	52273

Table 2: Counts for edge measures *interval degree*, *component degree* (for values from 1 to 3; larger values are not included), *level type* (for positive, nonpositive, and negative values), *level signature* (up to 10 most frequent values), and numbers of edges with *ancestor component roots* in their gaps and solely with ancestor component roots in their gaps; the second to last line gives the total numbers of non-projective edges, the last line gives the total numbers of all edges—we exclude edges from technical roots. (The listings need not be exhaustive; an empty cell means count zero.)

References

A. Abeillé, editor. 2003. *Treebanks: Building and Using Parsed Corpora*, volume 20 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht.

S. Afonso, E. Bick, R. Haber, and D. Santos. 2002. "Floresta sintá(c)tica": a treebank for Portuguese. In *Proceedings of the 3rd Intern. Conf. on Language Resources and Evaluation (LREC)*, pages 1698–1703.

Manuel Bodirsky, Marco Kuhlmann, and Matthias Möhl. 2005. Well-nested drawings as models of syntactic structure. In *Proceedings of Tenth Conference on Formal Grammar and Ninth Meering on Mathematics of Language*.

A. Böhmová, J. Hajič, E. Hajičová, and B. Hladká. 2003. The PDT: a 3-level annotation scenario. In Abeillé (2003), chapter 7.

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER treebank. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT)*.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*. SIGNLL.

M. Civit Torruella and M[a] A. Martí Antonín. 2002. Design principles for a Spanish treebank. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT)*.

Alexander Dikovsky and Larissa Modina. 2000. Dependencies on the other side of the Curtain. *Traitement Automatique des Langues (TAL)*, 41(1):67–96.

S. Džeroski, T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky, and A. Žele. 2006. Towards a Slovene dependency treebank. In *Proceedings of the 5th Intern. Conf. on Language Resources and Evaluation (LREC)*.

J. Hajič, O. Smrž, P. Zemánek, J. Šnaidauf, and E. Beška. 2004. Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*, pages 110–117.

Eva Hajičová, Jiří Havelka, Petr Sgall, Kateřina Veselá, and Daniel Zeman. 2004. Issues of Projectivity in the Prague Dependency Treebank. *Prague Bulletin of Mathematical Linguistics*, 81:5–22.

Jiří Havelka. 2005. Projectivity in Totally Ordered Rooted Trees: An Alternative Definition of Projectivity and Optimal Algorithms for Detecting Non-Projective Edges and Projectivizing Totally Ordered Rooted Trees. *Prague Bulletin of Mathematical Linguistics*, 84:13–30.

Jiří Havelka. 2007a. *Mathematical Properties of Dependency Trees and their Application to Natural Language Syntax*. Ph.D. thesis, Institute of Formal and Applied Linguistics, Charles University in Prague, Czech Republic.

Jiří Havelka. 2007b. Relationship between Non-Projective Edges, Their Level Types, and Well-Nestedness. In *Proceedings of HLT/NAACL; Companion Volume, Short Papers*, pages 61–64.

Tomáš Holan, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 1998. Two Useful Measures of Word Order Complexity. In Alain Polguère and Sylvain Kahane, editors, *Proceedings of Dependency-Based Grammars Workshop, COLING/ACL*, pages 21–28.

Tomáš Holan, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 2000. On Complexity of Word Order. *Traitement Automatique des Langues (TAL)*, 41(1):273–300.

Y. Kawata and J. Bartels. 2000. Stylebook for the Japanese treebank in VERBMOBIL. Verbmobil-Report 240, Seminar für Sprachwissenschaft, Universität Tübingen.

M. T. Kromann. 2003. The Danish dependency treebank and the underlying linguistic theory. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*.

Marco Kuhlmann and Joakim Nivre. 2006. Mildly Non-Projective Dependency Structures. In *Proceedings of COLING/ACL*, pages 507–514.

Solomon Marcus. 1965. Sur la notion de projectivité [On the notion of projectivity]. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 11:181–192.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of HLT/EMNLP*, pages 523–530.

Ladislav Nebeský. 1979. Graph theory and linguistics (chapter 12). In R. J. Wilson and L. W. Beineke, editors, *Applications of Graph Theory*, pages 357–380. Academic Press.

J. Nilsson, J. Hall, and J. Nivre. 2005. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. In *Proceedings of the NODALIDA Special Session on Treebanks*.

Joakim Nivre. 2006. Constraints on Non-Projective Dependency Parsing. In *Proceedings of EACL*, pages 73–80.

K. Oflazer, B. Say, D. Zeynep Hakkani-Tür, and G. Tür. 2003. Building a Turkish treebank. In Abeillé (2003), chapter 15.

K. Simov, P. Osenova, A. Simov, and M. Kouylekov. 2005. Design and implementation of the Bulgarian HPSG-based treebank. In *Journal of Research on Language and Computation – Special Issue*, pages 495–522. Kluwer Academic Publishers.

Neil J. A. Sloane. 2007. On-Line Encyclopedia of Integer Sequences. Published electronically at `www.research.att.com/~njas/sequences/`.

L. van der Beek, G. Bouma, R. Malouf, and G. van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*.

Kateřina Veselá, Jiří Havelka, and Eva Hajičová. 2004. Condition of Projectivity in the Underlying Dependency Structures. In *Proceedings of COLING*, pages 289–295.