Prague Dependency Treebank: Introduction

Markéta Lopatková, Jiří Mírovský
Institute of Formal and Applied Linguistics, MFF UK
lopatkova@ufal.mff.cuni.cz
NPFL075 Prague Dependency Treebank

Lectures:
  Markéta Lopatková  Fri, S8, 10:40-12:10 (cz/eng)
  Fri, S8, 14:00-15:30 (eng)

Practical sessions:
  Jiří Mírovský  Fri, SU1, 9:00-10:30

http://ufal.mff.cuni.cz/course/npfl075

Requirements:
  • Homework (35%)
  • Activity (15%)
  • Final test (50%)

Assessment:
  • excellent (= 1) > 90%
  • very good (= 2) > 70%
  • good (= 3) > 50%
Prague Dependency Treebank

Collection of:

- linguistically annotated data (Czech)
- tools and data format(s)
- documentation
Prague Dependency Treebank

Collection of:
- linguistically annotated data (Czech)
- tools and data format(s)
- documentation

Another point of view:
- annotation scheme
- framework for annotation of different languages
- underlying linguistic theory (Functional Generative Description)
Outline of the lecture

- trees (graph theory and data format)
- phrase structure trees and dependency trees
- dependency and non-dependency relations
- non-projectivity
How to capture sentence structure?
Graph theory: tree

tree (graph theory):

- finite graph \( \langle N, E \rangle \), \( N \)~nodes, \( E \)~edges/vertices \( \{n_1, n_2\} \)
- connected
- no cycles, no loops
- no more than 1 edge between any two different nodes

⇒ (undirected) graph

any two nodes are connected by exactly one simple path
Graph theory: tree

**tree** (graph theory):
- finite graph $\langle N, E \rangle$, $N$~nodes, $E$~edges/vertices $\{n_1, n_2\}$
- connected
- no cycles, no loops
- no more than 1 edge between any two different nodes

$\Rightarrow$ (undirected) graph
any two nodes are connected by exactly one simple path

**rooted tree**
- rooted $\Rightarrow$ orientation (i.e., edges ordered pairs $[n_1, n_2]$)
Graph theory: tree

**tree** (graph theory):
- finite graph \( \langle N, E \rangle \), \( N \sim \) nodes, \( E \sim \) edges/vertices \( \{n_1, n_2\} \)
- connected
- no cycles, no loops
- no more than 1 edge between any two different nodes

\( \Rightarrow \) (undirected) graph
any two nodes are connected by exactly one simple path

**rooted tree**
- rooted \( \Rightarrow \) orientation (i.e., edges ordered pairs \([n_1, n_2]\))

**directed tree** … directed graph
- which would be tree
  - if the directions on the edges were ignored, or
  - all edges are directed towards (or away from) a particular node \( \sim \) the **root**
Data structure: tree

tree as a data structure:

- finite directed graph \( \langle N, E \rangle \), \( N \sim \text{nodes}, E \sim \text{edges} \) \([n_1, n_2]\)
  - no cycles
  - connected
  - with root
- each non-root node has exactly one parent, and the root has no parent
  (each node has zero or more children nodes)
Data structure: tree

**tree as a data structure:**

- finite directed graph \( \langle N, E \rangle, N \text{--nodes}, E \text{--edges} [n_1, n_2] \)
  - no cycles
  - connected
  - with root
- each non-root node has exactly one parent, and the root has no parent
  (each node has zero or more children nodes)

+ (linear) ordering of nodes:
  the children of each node have a specific order
Data structure: tree (properties)

*tree as a data structure:*

- "tree-ordering" D … partial ordering on nodes
  \[ u \leq v \iff \text{the unique path from the root to } v \text{ passes through } u \]
  (weak ordering ~ reflexive, antisymmetric, transitive)

- "linear ordering" … (partial) ordering on nodes
  (strong ordering ~ antireflexive, asymmetric, transitive)
Tree-based structures in CL

two types of tree-based structures in CL:

- phrase structure tree / constituent structure tree
- dependency tree
Phrase structure tree

My brother often sleeps in his study.

Phrase structure tree (definition)

\[ T = \langle N, D, Q, P, L \rangle \]

\( \langle N, D \rangle \) … tree (as a data structure)

Q … lexical and grammatical categories

L … labeling function \( N \rightarrow Q \)

D … oriented edges ~ relation on lex. and gram. categories 

\textit{dominance relation}

+ 

P … relation on \( N \sim \) (partial strong linear ordering) 

\textit{precedence}
Phrase structure tree (definition)

\[ T = \langle N, D, Q, P, L \rangle \]

\langle N, D \rangle \ldots \textbf{tree} (as a data structure)

Q \ldots \text{lexical and grammatical categories}

L \ldots \text{labeling function } N \rightarrow Q

D \ldots \text{oriented edges } \sim \text{ relation on lex. and gram. categories}

\textit{dominance relation}

P \ldots \text{relation on } N \sim \text{ (partial strong linear ordering)}

\textit{precedence relation}

Relating dominance and precedence relations:

- \textit{exclusivity} condition for D and P relations
- ‘\textit{nontangling}’ condition
Phrase structure tree (relation P)

- **exclusivity** condition for D and P relations

\[ \forall x,y \in N \text{ holds: } ( [x,y] \in P \lor [y,x] \in P ) \iff ( [x,y] \notin D \land [y,x] \notin D ) \]
Phrase structure tree (relation $P$)

- **exclusivity** condition for D and P relations
  \[
  \forall x,y \in N \text{ holds: } ( [x,y] \in P \lor [y,x] \in P ) \iff ([x,y] \notin D \land [y,x] \notin D)
  \]

- **‘nontangling’** condition
  \[
  \forall w,x,y,z \in N \text{ holds: } ( [w,x] \in P \land [w,y] \in D \land [x,z] \in D ) \implies ([y,z] \in P)
  \]
Phrase structure tree (relation P)

- **exclusivity** condition for D and P relations
  \[ \forall x,y \in N \text{ holds: } ( [x,y] \in P \lor [y,x] \in P ) \iff ( [x,y] \notin D \land [y,x] \notin D ) \]

- **‘nontangling’** condition
  \[ \forall w,x,y,z \in N \text{ holds: } ( [w,x] \in P \land [w,y] \in D \land [x,z] \in D ) \implies ( [y,z] \in P ) \]

\[ T = \langle N,D,Q,P,L \rangle \text{ phrase structure tree} \]

- \[ \forall x,y \in N \text{ siblings } \implies [x,y] \in P \]
- the set of its leaves is totally ordered by P
Phrase structure tree

Pros

- derivation history / 'closeness' of a complementation
- coordination, apposition
- CFG-like
- derivation of a grammar
Phrase structure tree

derivation history / ‘closeness’:

…often sleeps in his study

... often sleeps in his study
Phrase structure tree

Pros
• derivation history / ‘closeness’ of a complementation
• coordination, apposition
• CFG-like
• derivation of a grammar

Contras
• complexity (number of non-terminal symbols)
• complement (‘two dependencies’) 
  příběhl bos
  [(he) arrived barefooted]
• free word order 
  discontinuous ‘phrases’
  non-projectivity
Po babiččině příjezdu půjdou rodiče do divadla.
[After grandma's arrival the parents will go to the theatre.]
discontinuous ‘phrases’: solution for English

Mary will eat bread.

What will Mary eat?

Phrase structure tree
discontinuous ‘phrases’: solution for English

Mary will eat bread.

What will Mary eat?
Corpora with phrase structure trees

- Penn Treebank (1995)
  http://www.cis.upenn.edu/~treebank/
  Penn Arabic Treebank, Penn Chinese Treebank
- International English Treebank (ICE)
  http://ice-corpora.net/ice/index.htm
- Paris 7
  http://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php
- Szeged Treebank 2.0
- many others
Dependency tree
Dependency tree

My brother often sleeps in his study.

Dependency tree (definition)

\[ T = \langle N, D, Q, WO, L \rangle \]

\langle N, D \rangle \ldots \textit{tree} (as a data structure)

Q \ldots \text{lexical and grammatical categories}

L \ldots \text{labeling function } N \rightarrow Q

D \ldots \text{oriented edges } \sim \text{ relation on lex. and gram. categories}

‘dependency’ relation

WO \ldots \text{relation on } N \sim \text{ (strong total ordering on } N) \ldots

word order

\[ \begin{align*}
\text{sleeps.Pred} & \quad \text{brother.Sb} \quad \text{study.Adv} \\
\text{my.Atr} & \quad \text{often.Adv} \quad \text{in.AuxP} \\
\text{his.Atr} & \end{align*} \]
**Pros**
- economical, clear
  (complex labels, ‘word’~ node)
- free word order
- head of a phrase

**Contras**
- no derivation history /
  'closeness'
- coordination, apposition
- complement
Po babiččině příjezdu půjdou rodiče do divadla.
[After grandma's arrival the parents will go to the theatre.]
Corpora with dependency trees

- PropBank (1995)
- family of Prague dependency treebanks: Czech, Arabic, English
- Danish Dep. Treebank
- Finnish: Turku Dependency Treebank
  [http://bionlp.utu.fi/fintreebank.html](http://bionlp.utu.fi/fintreebank.html)
- Negra corpus
- TIGERCorpus
  [http://www.ims.uni-stuttgart.de/projekte/TIGER/](http://www.ims.uni-stuttgart.de/projekte/TIGER/)
- SynTagRus Dependency Treebank for Russian
Dependency and non-dependency relations
Dependency and non-dependency relations

edges ~ *dependency relations* (prototypically)

- dependency relation: binary relation
- governing/modified unit (head) – dependent/modifyng unit (modifier)
- criterion: possible reduction
  
  ... dependent member of the pair may be deleted
  while the distributional properties are preserved (→ correctness is preserved)
Dependency and non-dependency relations

edges ~ **dependency relations** (prototypically)

- dependency relation: binary relation
- governing/modified unit (head) – dependent/modifying unit (modifier)
- criterion: possible reduction
  ... dependent member of the pair may be deleted while the distributional properties are preserved (→ correctness is preserved)

- endocentric constructions ... OK

*malý stůl* [small table], *přišel včas* [(he) came in time], *velmi brzo* [very soon]

- exocentric constructions ... **principle of analogy** on word classes

  *Prší.* [(It) rains.] ... ∃ subjectless verbs
  ⇒ *Král zemřel.* [The king died.] ... a verb rather than a noun is the head

  *The girl painted a bag.* → *The girl painted.* ... ∃ objectless verbs
  ⇒ *The girl carried a bag* ... an object is considered as depending on a verb
Dependency and non-dependency relations

BUT also other relations:

**coordination** … "multiplication" of a single syntactic position

- different referents
- coordination of sentence members / sentences
  
  *My sister Mary and John came late.*
  *Mary came in time but John was late.*
  *Nemohu odejít, neboť ještě nepřestalo pršet.*
  *[I can't leave since it hasn't stopped raining yet.]*

- coordination may be embedded
  
  *krásné a romantické hrady a zámky* [nice and romantic towers and castles]
Dependency and non-dependency relations

BUT also other relations:

**coordination** … "multiplication" of a single syntactic position

- different referents
- coordination of sentence members / sentences
  
  My sister Mary and John came late.
  Mary came in time but John was late.
  Nemohu odejít, neboť ještě nepřestalo pršet.
  [I can't leave since it hasn't stopped raining yet.]

- coordination may be embedded

  krásné a romantické hrady a zámky [nice and romantic towers and castles]

**apposition** … "multiplication" of a single syntactic position

- identical referent

  Charles IV, Holy Roman Emperor
  The Hobbit, or There and Back Again
Dependency and non-dependency relations

BUT also other relations:

**coordination** … "multiplication" of a single syntactic position

- different referents
- coordination of sentence members / sentences
  
  *My sister Mary and John came late.*
  *Mary came in time but John was late.*
  *Nemohu odejít, neboť ještě nepřestalo pršet.*
  *[I can't leave since it hasn't stopped raining yet.]*

- coordination may be embedded
  *krásné a romantické hrady a zámky* [nice and romantic towers and castles]

**apposition** … "multiplication" of a single syntactic position

- identical referent
  *Charles IV, Holy Roman Emperor*
  *The Hobbit, or There and Back Again*

→ necessary to enrich the data structure
Coordinating/apposition in dependency trees

PDT 2.0:

'connecting' constructions ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- connecting node (= node for coordinating / appositing conjunction)

[Diagram showing dependency tree with labeled nodes and edges.]
Coordination/apposition in dependency trees

PDT 2.0:

'connecting' constructions ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- **connecting node** (= node for coordinating / appositing conjunction)
- **effective parent** (= node for governing node, i.e. node modified by the whole construction, 'linguistic parent')

Diagram:

```
came  Coord
   /   
  Thin

youn Atr

Sb_Co men

Sb_Co soldiers
```
Coordination/apposition in dependency trees

PDT 2.0:

'connecting constructions' ~ coordination, apposition (, OPER)

specific types of nodes and edges:

- **connecting node** (= node for coordinating / appositing conjunction)
- **effective parent** (= node for governing node, i.e. node modified by the whole construction, 'linguistic parent')
- **members of a connecting construction** (= nodes that are coordinated / are in apposition)
  - is_member

```
came
Pred

men
Sb Co

soldiers
Sb Co

and
Coord

young
Atr

Thin
Atr
```
Coordination/apposition in dependency trees

PDT 2.0:

'connecting' constructions ~ coordination, apposition (, OPER)
n specific types of nodes and edges:

- **connecting node** (= node for coordinating / appositing conjunction)
- **effective parent** (= node for governing node, i.e. node modified by the whole construction, 'linguistic parent')
- **members of a connecting construction** (= nodes that are coordinated / are in apposition)
  - is_member
- **effective child(ren)** … modification(s) of the individual member of the connecting construction + common/shared modifier(s)
- ‘pass-through’ nodes
The center will gather and distribute the information on tenders and state commissions in this country as well as in abroad.
Coordination/apposition in dependency trees

PDT 2.0:
- embedded connecting constructions \[\rightarrow\] recursivity

- \textit{TrEd} (Tree Editor, Pajas):
  - functions \texttt{GetEChildren, GetEParents}
Coordination/apposition in dependency trees

Mel'čuk (1988):

‘grouping’ (G) … shared modification vs. modification of a single member

`Hubení ((mladí muži), vojáci a starci)`
[Thin young men, soldiers and old-men]
Dependency and non-dependency relations

other non-dependency relations in PDT:

• technical root – effective root of a sentence

• syntactically unclear expressions
  rhematizers; sentence, linking and modal adverbial expressions, conjunction modifiers

• list structures
  names, foreign expressions

• phrasemes
Projectivity and non-projectivity
A subtree $S$ of a rooted dependency tree $T$ is **projective** iff for all nodes $a$, $b$ and $c$ of the subtree $S$ the condition holds:

(1) $(a \leq_D b) \& (a <_{WO} b) \& (b \leq_{D^*} c) \Rightarrow (a <_{WO} c)$

and

(2) $(a \leq_D b) \& (b <_{WO} a) \& (b \leq_{D^*} c) \Rightarrow (c <_{WO} a)$
Projectivity and free word order

free word order:

- freedom of word order of dependents within a **continuous** ‘head domain’ (i.e., substring of head + its dependents)
- relaxation of continuity of a head domain

German:
Maria hat einen Mann kennengelernt der Schmetterlinge sammelt.
Mary has a man met who butteries collects
Mary has met a man who collects butteries
Projectivity and free word order

English: long-distance unbounded dependency

*John, Peter thought that Sue said that Mary loves.*
Projectivity and free word order

Czech:
Marii se Petr tu knihu rozhodl nekoupit.
to-Mary PART Peter that book decided not-buy
[Peter decided not to buy that book to Mary.]
Projectivity and non-projectivity

Projective dependency trees can be encoded by *linearization*:

- string of nodes, edges ~ brackets

$$\begin{align*}
\text{with WO ordering:} & \quad A \ ( B \ C \ ( D ) ) \\
\text{without WO ordering:} & \quad ( B ) \ A \ ( ( D ) \ C \ ( E ) )
\end{align*}$$
Projectivity and non-projectivity

Projective dependency trees can be encoded by **linearization**:

- string of nodes, edges ~ brackets

\[
A ( B C ( D ) ) \quad \text{without WO ordering}
\]
\[
( B ) A ( ( D ) C ( E ) ) \quad \text{with WO}
\]

\[
A ( B C ( D E F ( G ) ) ) ) \quad \text{without WO}
\]
\[
( B ) A ( C ( ( E ) D ( ( G ) F ) ) ) \quad \text{with WO}
\]
Planarity

A dependency graph $T$ is \textit{planar}, if it does not contain nodes $a$, $b$, $c$, $d$ such that:

$\text{linked}(a,c) \land \text{linked}(b,d) \land a <_{WO} b <_{WO} c <_{WO} d$

\textit{linked}(i,j) … ‘there is an edge in $T$ from $i$ to $j$, or vice versa’

My brother often sleeps in his study.
Planarity

A dependency graph $T$ is planar, if it does not contain nodes $a, b, c, d$ such that:

$$\text{linked}(a, c) \ & \ & \text{linked}(b, d) \ & \ & a <_\text{WO} b <_\text{WO} c <_\text{WO} d$$

$\text{linked}(i, j)$ … ‘there is an edge in $T$ from $i$ to $j$, or vice versa’

My brother often sleeps in his study.

Jan viděl větší město než Praha.
Planarity

A dependency graph $T$ is **planar**, if it does not contain nodes $a, b, c, d$ such that:

$\text{linked}(a,c)$ & $\text{linked}(b,d)$ & $a <_{\text{WO}} b <_{\text{WO}} c <_{\text{WO}} d$

$\text{linked}(i,j)$ … ‘there is an edge in $T$ from $i$ to $j$, or vice versa’

Informally, a dependency graph is planar, if its edges can be drawn above the sentence without crossing.

**Jan viděl větší město než Praha.**

**My brother often sleeps in his study.**
‘Well-Nestedness’

Two subtrees $T_1, T_2$ **interleave**, if there are nodes $l_1, r_1 \in T_1$ and $l_2, r_2 \in T_2$ such that

$$l_1 \prec_{WO} l_2 \prec_{WO} r_1 \prec_{WO} r_2$$

A dependency graph is **well-nested**, if no two of its disjoint subtrees interleave.'
‘Well-Nestedness’

Two subtrees $T_1, T_2$ *interleave*, if there are nodes $l_1, r_1 \in T_1$ and $l_2, r_2 \in T_2$ such that

$$l_1 <_{wo} l_2 <_{wo} r_1 <_{wo} r_2$$

A dependency graph is *well-nested*, if no two of its disjoint subtrees interleave.’
Gap Degree \( dNh(T) \)

**Coverage** of a node \( u \in T \)

\[
\text{Cov}(u, T) = \{ i \mid \text{i - word order position of } v \in T \text{ such that, } u \leq_D v \}
\]

\[
\begin{align*}
\text{Cov}(u_1, T) &= \{1\} \\
\text{Cov}(u_2, T) &= \{2\} \\
\text{Cov}(u_3, T) &= \{1, 2, 3, 4, 5\} \\
\text{Cov}(u_4, T) &= \{4\} \\
\text{Cov}(u_5, T) &= \{1, 5\}
\end{align*}
\]
**Gap Degree** $dNh(T)$

**Coverage** of a node $u \in T$

$Cov(u, T) = \{i \mid i - \text{word order position of } v \in T \text{ such that, } u \leq_D v\}$

**Gap in Coverage** of a node $u \in T \iff_{\text{def}} Cov(u, T)$ is not an interval

$Cov(u_1, T) = \{1\} \quad Cov(u_2, T) = \{2\} \quad Cov(u_3, T) = \{1, 2, 3, 4, 5\} \quad Cov(u_4, T) = \{4\} \quad Cov(u_5, T) = \{1, 5\}$
**Gap Degree** \( dN_{h}(T) \)

**Coverage** of a node \( u \in T \)

\[
\text{Cov}(u, T) = \{ i \mid i \text{ - word order position of } v \in T \text{ such that, } u \leq_D v \}
\]

**Gap in Coverage** of a node \( u \in T \) \( \Leftrightarrow \text{def} \) \( \text{Cov}(u, T) \) is not an interval

---

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Gap Degree $dNh(T)$

**Coverage** of a node $u \in T$

$Cov(u, T) = \{ i \mid i$ - word order position of $v \in T$ such that, $u \leq_D v \}$

**Gap in Coverage** of a node $u \in T$ $\iff$ _def_ $Cov(u, T)$ is not an interval

$dNh(u, T)$ … **number of Gaps** in $Cov(u, T)$

**Tree Degree** $dNh(T) = \max \{ dNh(u, T) \mid u \in T \}$

$Cov(u_1, T) = \{ 1 \}$  $Cov(u_2, T) = \{ 2 \}$  $Cov(u_3, T) = \{ 1, 2, 3, 4, 5 \}$  $Cov(u_4, T) = \{ 4 \}$  $Cov(u_5, T) = \{ 1, 5 \}$

![Tree Diagram]

- [našli, 3, 0]
  - [rodiče, 2, 3]
  - [synovì, 4, 3]
  - [nevěstu, 5, 3]
- [bohatou, 1, 5]
Gap Degree $dNh(T)$

Coverage of a node $u \in T$

$\text{Cov}(u, T) = \{ i \mid i -$ word order position of $v \in T$ such that, $u \preceq_D v \}$

Gap in Coverage of a node $u \in T \iff \text{def Cov}(u, T)$ is not an interval

$dNh(u, T) \ldots$ number of Gaps in $\text{Cov}(u, T)$

Tree Degree $dNh(T) = \max \{ dNh(u, T) \mid u \in T \}$
English: long-distance unbounded dependency
*John, Peter thought that Sue said that Mary loves.*
Projectivity and free word order

Czech:
Marii se Petr tu knihu rozhodl nekoupit.
to-Mary PART Peter that book decided not-buy
[Peter decided not to buy that book to Mary.]
Edge Degree

Let \( T = (N, E) \) dependency tree, \( e = [i, j] \) an edge in \( E \), \( T_e \) the subgraph of \( T \) induced by the nodes contained in the span of \( e \).

*Degree of an edge* \( e \in E \), \( ed(e) \), is the number of connected components \( c \) in \( T_e \) such that the root of \( c \) is not dominated by the head of \( e \).

*Edge degree of \( T \), \( ed(T) \) … \( \max \{ ed(e) | e \in T \} \)
Edge Degree

Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$, $T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.

**Degree of an edge** $e \in E$, $ed(e)$, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.

**Edge degree of $T$, $ed(T)$** ... $\max \{ed(e) | e \in T\}$
Edge Degree

Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$, $T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.

**Degree of an edge** $e \in E$, $ed(e)$, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.

**Edge degree of T, ed(T)** … $\max \{ed(e) | e \in T \}$
Edge Degree

Let $T = (N, E)$ dependency tree, $e = [i, j]$ an edge in $E$, $T_e$ the subgraph of $T$ induced by the nodes contained in the span of $e$.

**Degree of an edge** $e \in E$, $ed(e)$, is the number of connected components $c$ in $T_e$ such that the root of $c$ is not dominated by the head of $e$.

**Edge degree of $T$, $ed(T)$** $\ldots$ $\max \{ ed(e) | e \in T \}$
<table>
<thead>
<tr>
<th>property</th>
<th>DDT</th>
<th>PDT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n = 4393$</td>
<td>$n = 73088$</td>
</tr>
<tr>
<td>all structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gap degree 0</td>
<td>3732</td>
<td>56168</td>
</tr>
<tr>
<td></td>
<td>84.95%</td>
<td>76.85%</td>
</tr>
<tr>
<td>gap degree 1</td>
<td>654</td>
<td>16608</td>
</tr>
<tr>
<td></td>
<td>14.89%</td>
<td>22.72%</td>
</tr>
<tr>
<td>gap degree 2</td>
<td>7</td>
<td>307</td>
</tr>
<tr>
<td></td>
<td>0.16%</td>
<td>0.42%</td>
</tr>
<tr>
<td>gap degree 3</td>
<td>–</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>–</td>
<td>0.01%</td>
</tr>
<tr>
<td>gap degree 4</td>
<td>–</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>&lt; 0.01%</td>
<td></td>
</tr>
<tr>
<td>edge degree 0</td>
<td>3732</td>
<td>56168</td>
</tr>
<tr>
<td></td>
<td>84.95%</td>
<td>76.85%</td>
</tr>
<tr>
<td>edge degree 1</td>
<td>584</td>
<td>16585</td>
</tr>
<tr>
<td></td>
<td>13.29%</td>
<td>22.69%</td>
</tr>
<tr>
<td>edge degree 2</td>
<td>58</td>
<td>259</td>
</tr>
<tr>
<td></td>
<td>1.32%</td>
<td>0.35%</td>
</tr>
<tr>
<td>edge degree 3</td>
<td>17</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>0.39%</td>
<td>0.09%</td>
</tr>
<tr>
<td>edge degree 4</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>0.05%</td>
<td>0.01%</td>
</tr>
<tr>
<td>edge degree 5</td>
<td>–</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>&lt; 0.01%</td>
<td></td>
</tr>
<tr>
<td>edge degree 6</td>
<td>–</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>&lt; 0.01%</td>
<td></td>
</tr>
<tr>
<td>projective</td>
<td>3732</td>
<td>56168</td>
</tr>
<tr>
<td></td>
<td>84.95%</td>
<td>76.85%</td>
</tr>
<tr>
<td>planar</td>
<td>3796</td>
<td>60048</td>
</tr>
<tr>
<td></td>
<td>86.41%</td>
<td>82.16%</td>
</tr>
<tr>
<td>well-nested</td>
<td>4388</td>
<td>73010</td>
</tr>
<tr>
<td></td>
<td>99.89%</td>
<td>99.89%</td>
</tr>
<tr>
<td>non-projective structures only</td>
<td>$n = 661$</td>
<td>$n = 16920$</td>
</tr>
<tr>
<td>planar</td>
<td>64</td>
<td>3880</td>
</tr>
<tr>
<td></td>
<td>9.68%</td>
<td>22.93%</td>
</tr>
<tr>
<td>well-nested</td>
<td>656</td>
<td>16842</td>
</tr>
<tr>
<td></td>
<td>99.24%</td>
<td>99.54%</td>
</tr>
</tbody>
</table>
References


Coordination/apposition in dependency trees

Petkevič (1995) … formal representation of FGD

\[\text{cop} \quad \text{ACT} \quad \text{PAT} \quad \text{RSTR} \quad \text{RSTR} \quad \text{ACT} \quad \text{LOC} \quad \text{RSTR} \quad \text{PAT}\]

\[\langle[(\text{Jan},t); (\text{Marie},t)]_{\text{cop}} \quad \text{RSTR} \langle\langle(\text{ který},t)\rangle\rangle_{\text{ACT}} \quad (\text{žít},t) \quad \text{LOC} \langle\langle(\text{Boston},t)\rangle\rangle \rangle_{\text{ACT}} \quad (\text{být},f)\]

\[\text{PAT} \langle\langle(\text{dobrý},f)\rangle\rangle_{\text{RSTR}} \quad (\text{člověk},f)\]