Dependency Grammars and Treebanks: Intro – trees, word order, projectivity

Markéta Lopatková, Daniel Zeman, Jiří Mírovský

Institute of Formal and Applied Linguistics, MFF UK

lopatkova@ufal.mff.cuni.cz
Dependency Grammars and Treebanks (NPFL075)

Lectures: Wednesday, room S1, 15:40-17:10
Markéta Lopatková, Daniel Zeman

Practical sessions:
Jiří Mírovský, Daniel Zeman

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

Requirements:
• Homework (40%)
• Activity (10%)
• Final test (50%)

Assessment:
• excellent (= 1)  ≥ 90%
• very good (= 2)  ≥ 70%
• good (= 3)       ≥ 50%
Dependency Grammars and Treebanks

- Family of Prague Dependency Treebanks (PDT, PCEDT)
- Universal Dependencies
- HamleDT, PropBank, ???

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

Another point of view:
- annotation scheme
- framework for annotation of different languages
- underlying linguistic theory
How to capture sentence structure?

wsj_1411.treex.gz (64/108)
Gate **receipts** are only the Cowboys' second largest source of cash.
Graph theory: tree

tree (graph theory):

definition:

- finite graph \( \langle N, E \rangle \), \( N \sim \) nodes/vertices, \( E \sim \) edges \( \{n_1, n_2\} \)
- 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

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 a particular node \( \sim \) the root
Data structure: tree

**tree as a data structure:**
- rooted tree (as in graph theory)
- all edges are directed from a particular node ~ the **root**
- (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
  \[ u \leq v \overset{\text{def}}{\iff} \text{the unique path from the root to } v \text{ passes through } u \]
  (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
My brother often sleeps in his study.

Phrase structure tree (definition)

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

\langle N, D \rangle \ldots \textit{rooted tree, directed}

Q \ldots \text{lexical and grammatical categories}

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

D \ldots \text{oriented edges (branches)}

\sim \text{ relation on lex. and gram. categories}

\textit{dominance relation}

+ 

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

\text{relation of } \textit{precedence}
Phrase structure tree (definition)

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

\( \langle N, D \rangle \) … rooted tree, directed

Q … lexical and grammatical categories

L … labeling function \( N \to Q \)

D … oriented edges (branches)

~ relation on lex. and gram. categories

dominance relation

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

relation of precedence

Relating dominance and precedence relations:

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

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

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

- **exclusivity** condition for D and P relations
  \[ \forall x, y \in N \text{ holds: } ( [x, y] \in P \lor [y, x] \in P ) \iff ( [x, y] \not\in D \land [y, x] \not\in D) \]

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

\[ T = \langle N,D,Q, P,L \rangle \text{ phrase structure tree} \]
- \[ \forall x,y \in N \text{ siblings } \Rightarrow [x,y] \in P \]
- 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

Dependency Grammars and Treebanks - Intro
Phrase structure tree

derivation history / ‘closeness’:

…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’)
- free word order
- discontinuous ‘phrases’
- non-projectivity

přiběhl bos [(he) arrived barefooted]
Phrase structure tree

discontinuous ‘phrases’: solution for English

Mary will eat bread.

What will Mary eat?

Dependency Grammars and Treebanks - Intro
discontinuous ‘phrases’: solution for English

Mary will eat bread. What will Mary eat?

Dependency Grammars and Treebanks - Intro
Phrase structure tree

discontinuous ‘phrases’: solution for English

Mary will eat bread.

What will Mary eat?

T'  
   / \  
  NP  V'  
   |   |   
  what AuxV  
      /   |   
     NP  V  
        |   |   
       what eat bread

NP  
  |  
 N  
  |  
 Mary

NP  
  |  
 N  
  |  
 will eat bread

Dependency Grammars and Treebanks - Intro
Po babiččině příjezdu půjdou rodiče do divadla.

[After grandma's arrival
the parents will go to the theatre.]
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{rooted tree, directed}

\( Q \ldots \) lexical and grammatical categories

\( L \ldots \) labeling function \( N \rightarrow Q^+ \)

\( D \ldots \) oriented edges \( \sim \) relation on lex. and gram. categories

\textit{‘dependency’ relation}

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

\textit{word order}
Pros
- economical, clear (complex labels, ‘word’~ node)
- **free word order**
- head of a phrase

Contras
- no derivation history / 'closeness'
- **coordination**, apposition
- complement
Mary will eat bread.

What will Mary eat?

discontinuous ‘phrases’: no problem
Po babiččině příjezdu půjdou rodiče do divadla.
[After grandma's arrival the parents will go to the theatre.]
Projectivity and non-projectivity (definition)

Mark decided to marry Ann.

Nepodařilo se mi otevřít soubor.
Projectivity and non-projectivity (definition)

Whom did Mark decide to marry?

Soubor se mi nepodařilo otevřít. (Oliva)

\[
\text{decided.} \text{Pred} \\
\text{did.} \text{AuxV} \quad \text{Mark.} \text{Sb} \quad \text{to marry.} \text{Obj} \\
\text{Whom.} \text{Obj} \\
\]

\[
\text{nepodařilo.} \text{Pred} \\
\text{se.} \text{AuxT} \quad \text{mi.} \text{Obj} \quad \text{otevřít.} \text{Obj} \\
\text{Soubor.} \text{Obj} \\
\]
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. $(a \leq_D b) \& (a <_{WO} c <_{WO} b) \Rightarrow (a <_{D^*} c)$

and

2. $(a \leq_D b) \& (b <_{WO} c <_{WO} a) \Rightarrow (a <_{WO^*} c)$
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 the butterflies collects
‘Mary has met a man who collects butteries.’

English: long-distance unbounded dependency
John, Peter thought that Sue said that Mary loves.

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

```
( B ) A ( ( D ) C ( E ) )   with WO ordering

( B ) A ( C ( ( E ) D ( ( G ) F ) ) )   with WO ordering
```

```
A ( B C ( D E ) )   without WO ordering

( B ) A ( ( D ) C ( E ) )   with WO

A ( B C ( D ( E F ( G ) ) ) )   without WO ordering

( B ) A ( C ( ( E ) D ( ( G ) F ) ) )   with WO ordering
```
Planarity

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

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

`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.

My brother often sleeps in his study.

Jan viděl větší město než Praha.
Planarity vs. projectivity

projectivity $\Rightarrow$ planarity
projectivity $\not\Leftrightarrow$ planarity

(Kuhlmann, M., Nivre, J., 2006)
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.]
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.'
Planarity vs. projectivity

projectivity $\implies$ planarity $\implies$ well-nestedness

projectivity $\not\iff$ planarity $\not\iff$ well-nestedness

(Kuhlmann, M., Nivre, J., 2006)
**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, $u \leq_D v \}$

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

Diagram:
- decided, 4
  - did, 2
  - he, 3
- [to mary, 5]
- [whom, 1]
**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 $Cov(u, T)$

$Cov(u_1, T) = \{ 1 \}$; $Cov(u_2, T) = \{ 2 \}$; $Cov(u_3, T) = \{ 3 \}$; $Cov(u_4, T) = \{ 1, 2, 3, 4, 5 \}$; $Cov(u_5, T) = \{ 1, 5 \}$
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 Gap Degree** $dNh(T) = \max \{ dNh(u, T) \mid u \in T \}$

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

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

**Tree Gap Degree** $gd(T) = \max \{ gd(u, T) \mid u \in T \}$
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** $e \in E$, $\text{ed}(e)$, 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 $T$, $\text{ed}(T)$** … $\max \{\text{ed}(e) | e \in T \}$
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 $T$, $ed(T)$** … $\max \{ed(e) | e \in T \}$
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** $e \in E$, $\text{ed}(e)$, 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 $T$, $\text{ed}(T)$** $\cdots \max \{\text{ed}(e) | e \in T \}$
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** $e \in E$, $\text{ed}(e)$, 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 $T$, $\text{ed}(T)$** … max $\{\text{ed}(e) | e \in T\}$
Planarity vs. projectivity

projectivity $\Rightarrow$ planarity $\Rightarrow$ well-nestedness

projectivity $\not\iff$ planarity $\not\iff$ well-nestedness

$gd(T) = 0 \iff ed(T) = 0 \iff$ projectivity

well-nestedness … independent from gap/edge degree

\[ \forall \, d > 0 \text{ well-nested and non-well-nested trees exist such that } gd(T) = d \text{ and } ed(T) = d \]

(Kuhlmann, M., Nivre, J., 2006)
<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 84.95%</td>
<td>56168 76.85%</td>
</tr>
<tr>
<td>gap degree 1</td>
<td>654  14.89%</td>
<td>16608 22.72%</td>
</tr>
<tr>
<td>gap degree 2</td>
<td>7   0.16%</td>
<td>307  0.42%</td>
</tr>
<tr>
<td>gap degree 3</td>
<td>–   –</td>
<td>4   0.01%</td>
</tr>
<tr>
<td>gap degree 4</td>
<td>–   –</td>
<td>1   &lt; 0.01%</td>
</tr>
<tr>
<td>edge degree 0</td>
<td>3732 84.95%</td>
<td>56168 76.85%</td>
</tr>
<tr>
<td>edge degree 1</td>
<td>584  13.29%</td>
<td>16585 22.69%</td>
</tr>
<tr>
<td>edge degree 2</td>
<td>58   1.32%</td>
<td>259  0.35%</td>
</tr>
<tr>
<td>edge degree 3</td>
<td>17   0.39%</td>
<td>63   0.09%</td>
</tr>
<tr>
<td>edge degree 4</td>
<td>2    0.05%</td>
<td>10   0.01%</td>
</tr>
<tr>
<td>edge degree 5</td>
<td>–    –</td>
<td>2    &lt; 0.01%</td>
</tr>
<tr>
<td>edge degree 6</td>
<td>–    –</td>
<td>1    &lt; 0.01%</td>
</tr>
<tr>
<td>projective</td>
<td>3732 84.95%</td>
<td>56168 76.85%</td>
</tr>
<tr>
<td>planar</td>
<td>3796 86.41%</td>
<td>60048 82.16%</td>
</tr>
<tr>
<td>well-nested</td>
<td>4388 99.89%</td>
<td>73010 99.89%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>non-projective structures only</th>
<th>( n = 661 )</th>
<th>( n = 16920 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>planar</td>
<td>64 9.68%</td>
<td>3880 22.93%</td>
</tr>
<tr>
<td>well-nested</td>
<td>656 99.24%</td>
<td>16842 99.54%</td>
</tr>
</tbody>
</table>

Corpora with dependency trees

- PropBank (1995)
  http://propbank.github.io/
- family of Prague dependency treebanks: Czech, Arabic, English, …
  http://ufal.mff.cuni.cz/pdt.html
- HamleDT project (from 2012)
  http://ufal.mff.cuni.cz/hamledt
- **Universal Dependencies** (from 2013)
  http://universaldependencies.org/
- Danish Dep. Treebank
  http://mbkromann.github.io/copenhagen-dependency-treebank/
- Finnish: Turku Dependency Treebank
  http://bionlp.utu.fi/fintreebank.html
- Negra corpus
  http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/negra-corpus.html
- TIGERCorpus
  http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html/
- SynTagRus Dependency Treebank for Russian
References

A subtree $S$ of a rooted dependency tree $T$ is \textit{projective} iff for all nodes $a$, $b$ and $c$ of the subtree $S$ the condition holds:

\begin{enumerate}
\item $(a \leq_D b) \land (a <_{WO} b) \land (b \leq_{D^*} c) \Rightarrow (a <_{WO} c)$
\item $(a \leq_D b) \land (b <_{WO} a) \land (b \leq_{D^*} c) \Rightarrow (c <_{WO} a)$
\end{enumerate}

\textbf{counter-example:}