Prague Dependency Treebank: Introduction – (Non-)Projectivity

Markéta Lopatková, Jiří Mírovský

Institute of Formal and Applied Linguistics, MFF UK

lopatkova@ufal.mff.cuni.cz
Projectivity and non-projectivity (definition)

Mark decided to marry Ann.

\[
\text{decided}.\text{Pred} \\
\text{Mark}.\text{Sb} \quad \text{to marry}.\text{Obj} \\
\quad \text{Ann}.\text{Obj}
\]

Nepodařilo se mi otevřít soubor.

\[
\text{Nepodařilo}.\text{Pred} \\
\text{se}.\text{AuxT} \quad \text{mi}.\text{Obj} \\
\quad \text{otevřít}.\text{Obj} \\
\quad \text{soubor}.\text{Obj}
\]
Projectivity and non-projectivity (definition)

Whom did Mark decided to marry?  

*Soubor se mi nepodařilo otevřít.* (Oliva)
Projectivity and non-projectivity (definition)

Whom did Mark decide to marry?

Soubor se mi nepodařilo otevřít. (Oliva)
Projectivity and non-projectivity (definition)

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. \[ (a \leq_D b) \land (a <_{WO} b) \land (b \leq_{D^*} c) \implies (a <_{WO} c) \]

and

2. \[ (a \leq_D b) \land (b <_{WO} a) \land (b \leq_{D^*} c) \implies (c <_{WO} a) \]
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)
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:
María hat einen Mann kennengelernt der Schmetterlinge sammelt.
Mary has met a man who collects butterflies
Projectivity and free word order

English: long-distance unbounded dependency

*John, Peter thought that Sue said that Mary loves.*
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

\[
\begin{align*}
A & \quad (B) \quad A \quad (D) \\
B \quad C & \quad (B) \quad A \quad (D) \quad C \quad (E)
\end{align*}
\]

without WO ordering

with WO
Projectivity and non-projectivity

Projective dependency trees can be encoded by *linearization*:

- string of nodes, edges ~ brackets

- without WO ordering
  \[ A ( B C ( D E ) ) \]
  \[ ( B ) A ( ( D ) C ( E ) ) \]

- without WO ordering
  \[ A ( B C ( D ( E F ( G ) ) ) ) \]
  \[ ( B ) A ( C ( ( E ) D ( ( G ) F ) ) ) \]
Planarity

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

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

**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:

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

**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.

Jan - saw - bigger - city - than – Prague
Jan saw a city bigger than Prague.
Planarity

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

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

\textit{linked}(i,j) \ldots ‘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.
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-Nestededness’

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

$$l_1 <_W l_2 <_W r_1 <_W r_2$$

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

$$l_1 <_W O l_2 <_W O r_1 <_W O r_2$$

A dependency graph is **well-nested**, if no two of its disjoint subtrees interleave.”
**Gap Degree** \( d_{Nh}(T) \)

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

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

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

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

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

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

$\text{Cov}(u_1, T) = \{1\}; \quad \text{Cov}(u_2, T) = \{2\}; \quad \text{Cov}(u_3, T) = \{3\}; \quad \text{Cov}(u_4, T) = \{1, 2, 3, 4, 5\}; \quad \text{Cov}(u_5, T) = \{1, 5\}$
**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) \text{ is not an interval}

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

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

\[
\begin{align*}
Cov(u_1, T) &= \{1\}; \quad Cov(u_2, T) = \{2\}; \quad Cov(u_3, T) = \{3\}; \quad Cov(u_4, T) = \{1, 2, 3, 4, 5\}; \quad Cov(u_5, T) = \{1, 5\}\end{align*}
\]
**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 $u \in T \iff_{\text{def}} Cov(u, T)$ is not an interval

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

**Tree Degree** $dNh(T) = \max \{ dNh(u, T) \mid u \in T \}$
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** $e \in E$, $ed(e)$, 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 $T$, $ed(T)$** $\ldots \max \{ed(e)| e \in T\}$
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 $T$, ed($T$)** \( \ldots \max \{ \text{ed}(e) \mid e \in T \} \)
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** $e \in E$, $\text{ed}(e)$, 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 $T$, $\text{ed}(T)$** $\ldots \max \{\text{ed}(e) | e \in T\}$
<table>
<thead>
<tr>
<th>property</th>
<th>DDT</th>
<th>PDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>all structures</td>
<td>$n = 4393$</td>
<td>$n = 73088$</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>–</td>
<td>&lt; 0.01%</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>–</td>
<td>&lt; 0.01%</td>
</tr>
<tr>
<td>edge degree 6</td>
<td>–</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>–</td>
<td>&lt; 0.01%</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>
</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</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