# Sentence diagrams: their evaluation and combination

Jirka Hana Barbora Hladká Ivana Lukšová

Charles University in Prague,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
Prague, Czech Republic
http://ufal.mff.cuni.cz/capek

# Motivation

- Data: treebanks in HamleDT
- Annotation scheme: Prague Dependency Treebank style
- Parser: Malt Parser 1.7
- Performance measure: Unlabeled Attachment Score

| ar | bg | bn | ca | cs | da | de | el | en | es |
|---|---|---|---|---|---|---|---|---|---|
| 80.4 | 90.9 | 80.3 | 89.7 | 86.7 | 88.0 | 88.4 | 82.5 | 88.2 | 89.8 |

| et | eu | fa | fi | grc | hi | hu | it | ja | la |
|---|---|---|---|---|---|---|---|---|---|
| 88.9 | 80.7 | 84.1 | 80.3 | 62.9 | 94.0 | 81.5 | 83.1 | 90.2 | 53.0 |

| nl | pl | pt | ro | ru | sk | sl | sv | ta | te |
|---|---|---|---|---|---|---|---|---|---|
| 81.4 | 91.2 | 86.7 | 84.2 | 85.4 | 82.2 | 82.0 | 85.0 | 77.4 | 90.3 |

| tr | | **AVG** |
|---|---|---|
| 81.6 | | **83.6** |

Credit to Daniel Zeman.

# The more data the better

- The results are not that great.
- More data should help.
- Annotated data are expensive.

  → **Crowdsourcing**

# The more data the better

- The results are not that great.
- More data should help.
- Annotated data are expensive.

  → **Crowdsourcing**

# The more data the better

- The results are not that great.
- More data should help.
- Annotated data are expensive.

  → **Crowdsourcing**

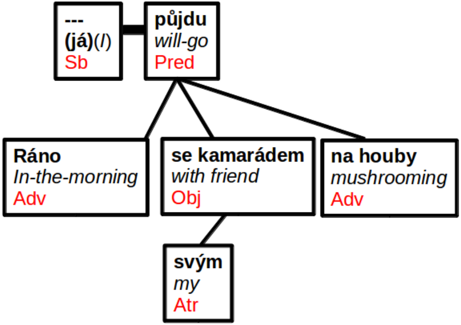
# The more data the better

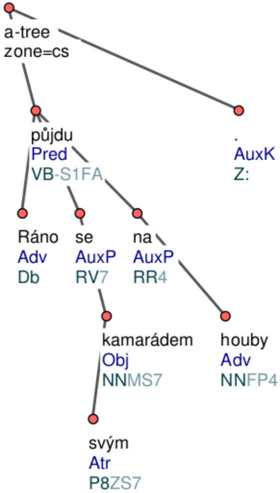
- The results are not that great.
- More data should help.
- Annotated data are expensive.

  → **Crowdsourcing**

# Sentence diagrams and treebanks

capture relationships between words in the sentence.





*I will go mushrooming with my friend in the morning.*

# Our goals

1. Collecting sentence diagrams produced by teachers and students.
    1. Design a tool for drawing sentence diagrams.
    2. Collect diagrams of suitable quality and quantity.
2. Using sentence diagrams as training data for parsers.

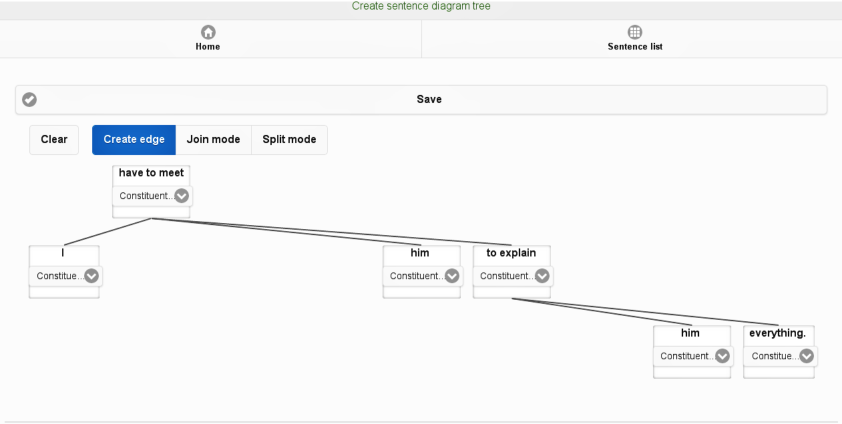
# Our goals

1. Collecting sentence diagrams produced by teachers and students.
   1. Design a tool for drawing sentence diagrams.
   2. Collect diagrams of suitable quality and quantity.
2. Using sentence diagrams as training data for parsers.

# Our goals

1. Collecting sentence diagrams produced by teachers and students.
    1. Design a tool for drawing sentence diagrams.
    2. Collect diagrams of suitable quality and quantity.
2. Using sentence diagrams as training data for parsers.

# Čapek: A tool for drawing sentence diagrams



http://capek.herokuapp.com/?lang=en → log in as 'guest', pwd: 'Guest1'

# Data quality

Two aspects

- Similarity between sentence diagrams
- Combination of multiple diagrams

# Similarity of sentence diagrams: Tree edit distance

- $D_1$, $D_2$ – two diagrams over an $n$-token sentence
- $TED(D_1, D_2, n)$ – the minimal cost of turning $D_2$ into $D_1$ using a set of simple operations; normalized by $n$; inspired by (Bille, 2005)

$$TED(D_1, D_2, n) = \min \frac{\#SPL + \#JOIN + \#INS + \#LINK + \#SLAB}{n}$$

# Similarity of sentence diagrams: Tree edit distance

- $D_1$, $D_2$ – two diagrams over an $n$-token sentence
- $TED(D_1, D_2, n)$ – the minimal cost of turning $D_2$ into $D_1$ using a set of simple operations; normalized by $n$; inspired by (Bille, 2005)

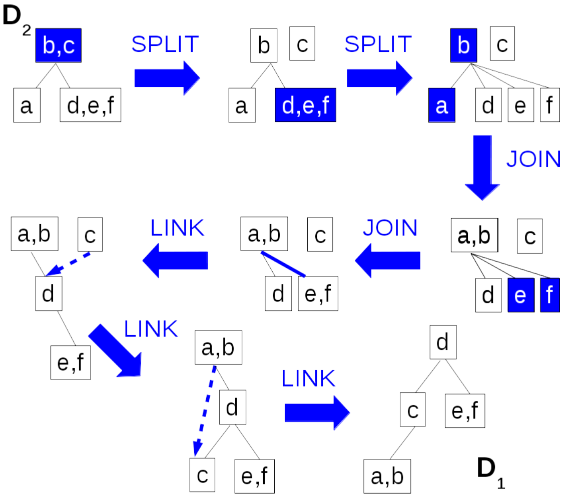
$$TED(D_1, D_2, n) = \min \frac{\#SPL + \#JOIN + \#INS + \#LINK + \#SLAB}{n}$$

# Similarity of sentence diagrams: Tree edit distance

- $D_1$, $D_2$ – two diagrams over an $n$-token sentence
- $TED(D_1, D_2, n)$ – the minimal cost of turning $D_2$ into $D_1$ using a set of simple operations; normalized by $n$; inspired by (Bille, 2005)

$$TED(D_1, D_2, n) = \min \frac{\#SPL + \#JOIN + \#INS + \#LINK + \#SLAB}{n}$$

# Similarity of sentence diagrams



$TED(D_1, D_2, 6) = 7/6$

# Combination of sentence diagrams

Goal: Combine $m$ diagrams $D_1, \ldots, D_m$ over a sentence $S = w_1 w_2 \ldots w_n$ into a single diagram by majority voting.

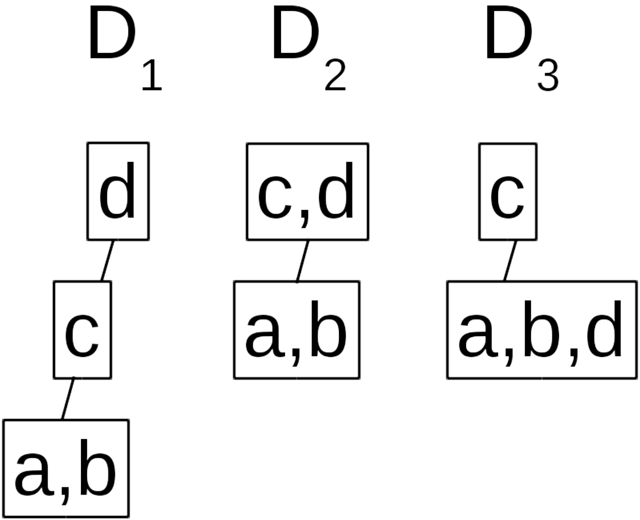
- First, determine the set of nodes (*FinalNodes*),
- Then, determine the set of edges (*FinalEdges*) over those nodes.

# Combination of sentence diagrams

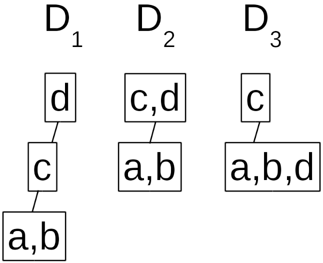
Goal: Combine $m$ diagrams $D_1, \ldots, D_m$ over a sentence $S = w_1 w_2 \ldots w_n$ into a single diagram by majority voting.

- First, determine the set of nodes (*FinalNodes*),
- Then, determine the set of edges (*FinalEdges*) over those nodes.

# Combination of sentence diagrams



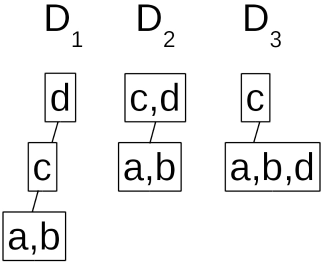
# Combination of sentence diagrams: Nodes



|  | *a* | *b* | *c* | *d* |
|---|---|---|---|---|
| *a* | x | **3** | 0 | 1 |
| *b* | x | x | 0 | 1 |
| *c* | x | x | x | 1 |
| *d* | x | x | x | x |

$FinalNodes = \{[a, b], [c], [d]\}$

# Combination of sentence diagrams: Nodes



| | *a* | *b* | *c* | *d* |
|---|---|---|---|---|
| *a* | x | **3** | 0 | 1 |
| *b* | x | x | 0 | 1 |
| *c* | x | x | x | 1 |
| *d* | x | x | x | x |

$FinalNodes = \{[a, b], [c], [d]\}$

# Combination of sentence diagrams: Edges

- $\mathit{FinalNodes} = \{[a, b], [c], [d]\}, \mathit{FinalEdges} = ?$

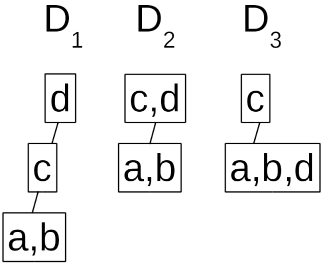
- Step 1: Assign weights to all token pairs (in each diagram)
- Step 2: Assign weights to all node pairs, i.e. potential edges
- Step 3: Greedily build a tree over the set of nodes.

# Combination of sentence diagrams: Edges

- $\mathit{FinalNodes} = \{[a,b],[c],[d]\}, \mathit{FinalEdges} =?$

- Step 1: Assign weights to all token pairs (in each diagram)
- Step 2: Assign weights to all node pairs, i.e. potential edges
- Step 3: Greedily build a tree over the set of nodes.

# Combination of sentence diagrams: Edges
## Step 1: Assign weights to all token pairs



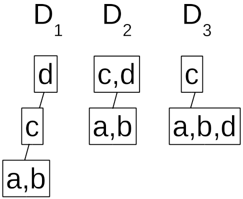
| token pair | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|
| $(a, b)$ | 0 | 0 | 0 |
| $(a, c)$ | 1/2 | 1/4 | 1/3 |
| $(a, d)$ | 0 | 1/4 | 0 |
| $(b, a)$ | 0 | 0 | 0 |
| $(b, c)$ | 1/2 | 1/4 | 1/3 |
| $(b, d)$ | 0 | 1/4 | 0 |
| $(c, a)$ | 0 | 0 | 0 |
| $(c, b)$ | 0 | 0 | 0 |
| $(c, d)$ | 1 | 0 | 0 |
| $(d, a)$ | 0 | 0 | 0 |
| $(d, b)$ | 0 | 0 | 0 |
| $(d, c)$ | 0 | 0 | 1/3 |

# Combination of sentence diagrams: Edges
# Step 2: Assign weights to all node pairs

$$\forall E = (N_1, N_2) \in Nodes \times Nodes :$$
$$weight(E) = \sum_{(t,u) \in tokens(N_1) \times \in tokens(N_2)} \sum_{d=1}^{m} \mathrm{weight}^d(t, u)$$

Weight of $([a, b], [c]) = (1/2+1/4+1/3) + (1/2+1/4+1/3) = 13/6$

Because:



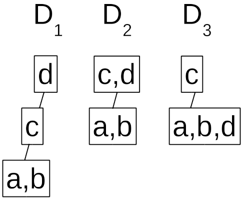
| token pair | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|
| $(a, c)$ | 1/2 | 1/4 | 1/3 |
| $(b, c)$ | 1/2 | 1/4 | 1/3 |

# Combination of sentence diagrams: Edges
# Step 2: Assign weights to all node pairs

$$\forall E = (N_1, N_2) \in \mathit{Nodes} \times \mathit{Nodes} :$$
$$\mathit{weight}(E) = \sum_{(t,u) \in \mathit{tokens}(N_1) \times \in \mathit{tokens}(N_2)} \sum_{d=1}^{m} \mathrm{weight}^d(t, u)$$

Weight of $([a, b], [c]) = (1/2+1/4+1/3) + (1/2+1/4+1/3) = 13/6$

Because:



| token pair | $D_1$ | $D_2$ | $D_3$ |
|---|---|---|---|
| $(a, c)$ | 1/2 | 1/4 | 1/3 |
| $(b, c)$ | 1/2 | 1/4 | 1/3 |

# Combination of sentence diagrams: *FinalEdges* Step 3: Greedily build a tree over the set of nodes.

$FinalNodes = \{[a, b], [c], [d]\}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | *FinalEdges* | | | | | | |
| | *PotentialEdges* | $([a, b], [c])$ | $([c], [d])$ | $([a, b], [d])$ | $([c], [a, b])$ | $([d], [a, b])$ | $([d], [c])$ |
| | weight | 13/6 | 1 | 1/2 | 0 | 0 | 0 |
| 1st | *FinalEdges* | $([a, b], [c])$ | | | | | |
| | *PotentialEdges* | | $([c], [d])$ | $([a, b], [d])$ | ~~$([c], [a, b])$~~ | $([d], [a, b])$ | $([d], [c])$ |
| 2nd | *FinalEdges* | $([a, b], [c])$ | $([c], [d])$ | | | | |
| | *PotentialEdges* | | | ~~$([a, b], [d])$~~ | | ~~$([d], [a, b])$~~ | ~~$([d], [c])$~~ |

- Thus: $FinalEdges = \{([a, b], [c]), ([c], [d])\}$

# Combination of sentence diagrams: *FinalEdges*
# Step 3: Greedily build a tree over the set of nodes.

$$FinalNodes = \{[a, b], [c], [d]\}$$

| | *FinalEdges* | | | | | | |
|---|---|---|---|---|---|---|---|
| | *PotentialEdges* | $([a, b], [c])$ | $([c], [d])$ | $([a, b], [d])$ | $([c], [a, b])$ | $([d], [a, b])$ | $([d], [c])$ |
| | weight | 13/6 | 1 | 1/2 | 0 | 0 | 0 |
| $1^{st}$ | *FinalEdges* | $([a, b], [c])$ | | | | | |
| | *PotentialEdges* | | $([c], [d])$ | $([a, b], [d])$ | ~~$([c], [a, b])$~~ | $([d], [a, b])$ | $([d], [c])$ |
| $2^{nd}$ | *FinalEdges* | $([a, b], [c])$ | $([c], [d])$ | | | | |
| | *PotentialEdges* | | | ~~$([a, b], [d])$~~ | | ~~$([d], [a, b])$~~ | ~~$([d], [c])$~~ |

- Thus: $FinalEdges = \{([a, b], [c]), ([c], [d])\}$

# Combination of sentence diagrams: *FinalEdges*
# Step 3: Greedily build a tree over the set of nodes.

$$FinalNodes = \{[a, b], [c], [d]\}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | *FinalEdges* | | | | | | |
| | *PotentialEdges* | $([a, b], [c])$ | $([c], [d])$ | $([a, b], [d])$ | $([c], [a, b])$ | $([d], [a, b])$ | $([d], [c])$ |
| | weight | 13/6 | 1 | 1/2 | 0 | 0 | 0 |
| 1st | *FinalEdges* | $([a, b], [c])$ | | | | | |
| | *PotentialEdges* | | $([c], [d])$ | $([a, b], [d])$ | ~~$([c], [a, b])$~~ | $([d], [a, b])$ | $([d], [c])$ |
| 2nd | *FinalEdges* | $([a, b], [c])$ | $([c], [d])$ | | | | |
| | *PotentialEdges* | | | ~~$([a, b], [d])$~~ | | ~~$([d], [a, b])$~~ | ~~$([d], [c])$~~ |

- Thus: $FinalEdges = \{([a, b], [c]), ([c], [d])\}$

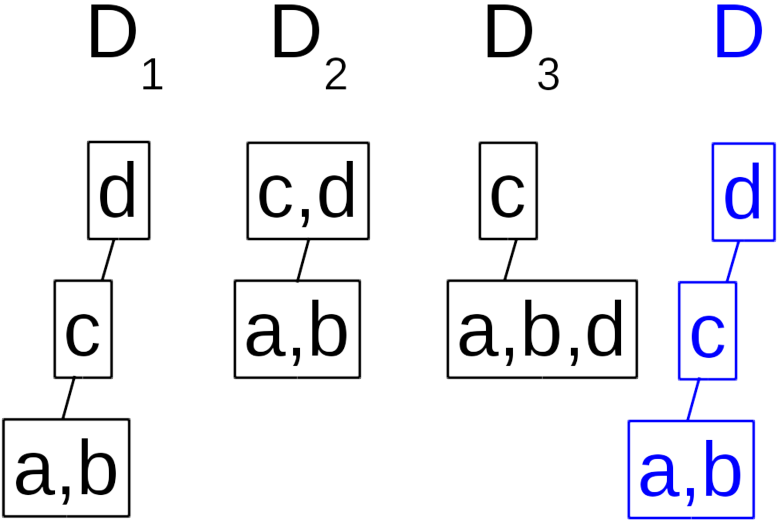
# Combination of sentence diagrams: *FinalEdges*
# Step 3: Greedily build a tree over the set of nodes.

$FinalNodes = \{[a, b], [c], [d]\}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | *FinalEdges* | | | | | | |
| | *PotentialEdges* | $([a, b], [c])$ | $([c], [d])$ | $([a, b], [d])$ | $([c], [a, b])$ | $([d], [a, b])$ | $([d], [c])$ |
| | weight | 13/6 | 1 | 1/2 | 0 | 0 | 0 |
| 1st | *FinalEdges* | $([a, b], [c])$ | | | | | |
| | *PotentialEdges* | | $([c], [d])$ | $([a, b], [d])$ | ~~$([c], [a, b])$~~ | $([d], [a, b])$ | $([d], [c])$ |
| 2nd | *FinalEdges* | $([a, b], [c])$ | $([c], [d])$ | | | | |
| | *PotentialEdges* | | | ~~$([a, b], [d])$~~ | | ~~$([d], [a, b])$~~ | ~~$([d], [c])$~~ |

- Thus: $FinalEdges = \{([a, b], [c]), ([c], [d])\}$

# Combination of sentence diagrams



# Sentence diagrams in Czech classes

- workbench of 101 sentences
- teachers ($T_1$, $T_2$), secondary school students ($S_1$, $S_2$), undergraduates ($U_1, \ldots, U_7$)

| | (T1,T2) | (T1,S1) | (T1,S2) | (S1,S2) |
|---|---|---|---|---|
| # of sentences | 101 | 91 | 101 | 91 |
| $\overline{TED}$ | 0.26 | 0.49 | 0.56 | 0.69 |

| | U1 | U2 | U3 | U4 | U5 | U6 | U7 | MV |
|---|---|---|---|---|---|---|---|---|
| # of sentences | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| $\overline{TED}$ | 0.78 | 0.63 | 0.56 | 0.76 | 0.38 | 0.62 | 1.21 | 0.40 |

(relative to T1)

# Sentence diagrams in Czech classes

- workbench of 101 sentences
- teachers ($T_1$, $T_2$), secondary school students ($S_1$, $S_2$), undergraduates ($U_1, \ldots, U_7$)

| | (T1,T2) | (T1,S1) | (T1,S2) | (S1,S2) |
|---|---|---|---|---|
| # of sentences | 101 | 91 | 101 | 91 |
| $\overline{TED}$ | 0.26 | 0.49 | 0.56 | 0.69 |

| | U1 | U2 | U3 | U4 | U5 | U6 | U7 | MV |
|---|---|---|---|---|---|---|---|---|
| # of sentences | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| $\overline{TED}$ | 0.78 | 0.63 | 0.56 | 0.76 | 0.38 | 0.62 | 1.21 | 0.40 |

(relative to T1)

# Sentence diagrams in Czech classes

- workbench of 101 sentences
- teachers ($T_1$, $T_2$), secondary school students ($S_1$, $S_2$), undergraduates ($U_1, \ldots, U_7$)

| | (T1,T2) | (T1,S1) | (T1,S2) | (S1,S2) |
|---|---|---|---|---|
| # of sentences | 101 | 91 | 101 | 91 |
| $\overline{TED}$ | 0.26 | 0.49 | 0.56 | 0.69 |

| | U1 | U2 | U3 | U4 | U5 | U6 | U7 | MV |
|---|---|---|---|---|---|---|---|---|
| # of sentences | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| $\overline{TED}$ | 0.78 | 0.63 | 0.56 | 0.76 | 0.38 | 0.62 | 1.21 | 0.40 |

(relative to T1)

# Thank you!

`http://capek.herokuapp.com/?lang=en` → log in as 'guest', pwd: 'Guest1'