Lingua-Align: An Experimental Toolbox for Automatic Tree-to-Tree Alignment
http://stp.lingfil.uu.se/~joerg/treealigner

Jörg Tiedemann

jorg.tiedemann@lingfil.uu.se
Department of Linguistics and Philology
Uppsala University

May 2010
Motivation

Aligning syntactic trees to create parallel treebanks

- phrase & rule extraction for (statistical) MT
- data for CAT, CALL applications
- corpus-based contrastive/translation studies

Framework:

- tree-to-tree alignment (automatically parsed corpora)
- classifier-based approach + alignment inference
- supervised learning using a rich feature set

→ Lingua::Align – feature extraction, alignment & evaluation
Example Training Data (SMULTRON)

1. predict individual links (local classifier)
2. align entire trees (global alignment inference)
Step 1: Link Prediction

- binary classifier
- log-linear model (MaxEnt)
- weighted feature functions $f_k$

\[
P(a_{ij}|s_i, t_j) = \frac{1}{Z(s_i, t_j)} \exp \left( \sum_k \lambda_k f_k(s_i, t_j, a_{ij}) \right)
\]

→ learning task: find optimal feature weights $\lambda_k$
Alignment Features

Feature engineering is important!

- real-valued & binary feature functions
- many possible features and feature combinations
- language-independent & language specific features
- directly from annotated corpora vs. features using additional resources
Alignment Features: Lexical Equivalence

Link score $\gamma$ based on probabilistic bilingual lexicons ($P(s_i|t_m)$ and $P(t_m|s_i)$ created by GIZA++):

$$\gamma(s, t) = \alpha(s|t)\alpha(t|s)\alpha(\overline{s}|\overline{t})\alpha(\overline{t}|\overline{s})$$

(Zhechev & Way, 2008)

**Idea:** Good links imply strong relations between tokens within subtrees to be aligned (*inside:* $\langle s; t \rangle$) & also strong relations between tokens outside of the subtrees to be aligned (*outside:* $\langle \overline{s}; \overline{t} \rangle$)
Alignment Features: Word Alignment

Based on (automatic) word alignment: How consistent is the proposed link with the underlying word alignments?

\[ \text{align}(s, t) = \frac{\sum_{L_{xy}} \text{consistent}(L_{xy}, s, t)}{\sum_{L_{xy}} \text{relevant}(L_{xy}, s, t)} \]

- \( \text{consistent}(L_{xy}, s, t) \): number of consistent word links
- \( \text{relevant}(L_{xy}, s, t) \): number of links involving tokens dominated by current nodes (relevant links)

\[ \Rightarrow \text{proportion of consistent links!} \]
Alignment Features: Other Base Features

- tree-level similarity (vertical position)
- tree-span similarity (horizontal position)
- nr-of-leaf-ratio (sub-tree size)
- POS/category label pairs (binary features)
Contextual Features

Tree alignment is structured prediction!

- local binary classifier: predictions in isolation
- implicit dependencies: include features from the context
- features of parent nodes, child nodes, sister nodes, grandparents ...

→ Lots of contextual features possible!
→ Can also create complex features!
Example Features

Some possible features for node pair $\langle DT_1, NN_3 \rangle$

<table>
<thead>
<tr>
<th>feature</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>labels=DT-NN</td>
<td>1</td>
</tr>
<tr>
<td>tree-span-similarity</td>
<td>0</td>
</tr>
<tr>
<td>tree-level-similarity</td>
<td>1</td>
</tr>
<tr>
<td>sister_labels=PP-NP</td>
<td>1</td>
</tr>
<tr>
<td>sister_labels=NNP-NP</td>
<td>1</td>
</tr>
<tr>
<td>parent$<em>{\alpha</em>{inside}}(t</td>
<td>s)$</td>
</tr>
<tr>
<td>srcparent_GIZA$_{src2trg}$</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Structured Prediction with History Features

- likelihood of a link depends on other link decisions
- for example: if parent nodes are linked, their children are also more likely to be linked (or not?)

→ **Link dependencies via history features:**

Children-link-feature: proportion of linked child-nodes
Subtree-link-feature: proportion of linked subtree-nodes
Neighbor-link-feature: binary link flag for left neighbors

→ **Bottom-up, left-to-right classification!**
Step 2: Alignment Inference

- use classification likelihoods as local link scores
- apply search procedure to align (all) nodes of both trees

→ global optimization as assignment problem
→ greedy alignment strategies
→ constrained link search

- many strategies/heuristics/combinations possible
- this step is optional (could just use classifier decisions)
Maximum weight matching

Apply graph-theoretic algorithms for “node assignment”

- aligned trees as weighted bipartite graphs
- assignment problem: matching with maximum weight

\[
Kuhn – Munkres \begin{bmatrix}
p_{11} & p_{12} & \cdots & p_{1n} \\
p_{21} & p_{22} & \cdots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \cdots & p_{nn}
\end{bmatrix} = \begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_n
\end{bmatrix}
\]

→ optimal one-to-one node alignment
Greedy Link Search

- greedy best-first strategy
- allow only one link per node
- = competitive linking strategy

Additional constraints: well-formedness (Zhechev & Way)
(no inconsistent links)

→ simple, fast, often optimal
→ easy to integrate important constraints
Some experiments

The TreeAligner requires training data!

- aligned parallel treebank: SMULTRON
  (http://www.ling.su.se/dali/research/smultron/index.htm)
- manual alignment
- Swedish-English (Swedish-German)
- 2 chapters of Sophie’s World (+ economical texts)
- 6,671 “good” links, 1,141 “fuzzy” links in about 500 sentence pairs

Train on 100 sentences from Sophie’s World (Swedish-English) (Test on remaining sentence pairs)
Evaluation

Precision = \frac{|P \cap A|}{|A|} \quad \text{Recall} = \frac{|S \cap A|}{|S|}

F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}

S = \text{sure ("good") links}
P = \text{possible ("fuzzy" + "good") links}
A = \text{links proposed by the system}
# Results on different feature sets (F-scores)

<table>
<thead>
<tr>
<th>inference → history</th>
<th>threshold=0.5</th>
<th>graph-assign</th>
<th>greedy</th>
<th>+wellformed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no</td>
<td>yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lexical</td>
<td>38.52</td>
<td>40.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ tree</td>
<td>50.27</td>
<td>51.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ alignment</td>
<td>60.41</td>
<td>60.63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ labels</td>
<td>72.44</td>
<td>72.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ context</td>
<td>74.68</td>
<td>74.90</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

→ **additional features always help**
Results on different feature sets (F-scores)

<table>
<thead>
<tr>
<th>inference →</th>
<th>threshold=0.5</th>
<th>graph-assign</th>
<th>greedy</th>
<th>+wellformed</th>
</tr>
</thead>
<tbody>
<tr>
<td>history →</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>lexical</td>
<td>38.52</td>
<td>40.00</td>
<td>49.75</td>
<td>56.60</td>
</tr>
<tr>
<td>+ tree</td>
<td>50.27</td>
<td>51.84</td>
<td>54.41</td>
<td>57.01</td>
</tr>
<tr>
<td>+ alignment</td>
<td>60.41</td>
<td>60.63</td>
<td>61.31</td>
<td>60.83</td>
</tr>
<tr>
<td>+ labels</td>
<td>72.44</td>
<td>72.24</td>
<td>72.72</td>
<td>73.05</td>
</tr>
<tr>
<td>+ context</td>
<td>74.68</td>
<td>74.90</td>
<td>74.96</td>
<td>75.38</td>
</tr>
</tbody>
</table>

→ additional features always help
→ alignment inference is important (with weak features)
## Results on different feature sets (F-scores)

<table>
<thead>
<tr>
<th>inference →</th>
<th>threshold=0.5</th>
<th>graph-assign</th>
<th>greedy</th>
<th>+wellformed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>lexical</td>
<td>38.52</td>
<td>40.00</td>
<td>49.75</td>
<td>56.60</td>
</tr>
<tr>
<td>+ tree</td>
<td>50.27</td>
<td>51.84</td>
<td>54.41</td>
<td>57.01</td>
</tr>
<tr>
<td>+ alignment</td>
<td>60.41</td>
<td>60.63</td>
<td>61.31</td>
<td>60.83</td>
</tr>
<tr>
<td>+ labels</td>
<td>72.44</td>
<td>72.24</td>
<td>72.72</td>
<td>73.05</td>
</tr>
<tr>
<td>+ context</td>
<td>74.68</td>
<td>74.90</td>
<td>74.96</td>
<td>75.38</td>
</tr>
</tbody>
</table>

→ additional features always help  
→ alignment inference is important (with weak features)  
→ greedy search is (at least) as good as graph-based assignment
### Results on different feature sets (F-scores)

<table>
<thead>
<tr>
<th>inference →</th>
<th>threshold=0.5</th>
<th>graph-assign</th>
<th>greedy</th>
<th>+wellformed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>lexical</td>
<td>38.52</td>
<td>40.00</td>
<td>49.75</td>
<td>56.60</td>
</tr>
<tr>
<td>+ tree</td>
<td>50.27</td>
<td>51.84</td>
<td>54.41</td>
<td>57.01</td>
</tr>
<tr>
<td>+ alignment</td>
<td>60.41</td>
<td>60.63</td>
<td>61.31</td>
<td>60.83</td>
</tr>
<tr>
<td>+ labels</td>
<td>72.44</td>
<td>72.24</td>
<td>72.72</td>
<td>73.05</td>
</tr>
<tr>
<td>+ context</td>
<td>74.68</td>
<td>74.90</td>
<td>74.96</td>
<td>75.38</td>
</tr>
</tbody>
</table>

→ **additional features always help**
→ **alignment inference is important (with weak features)**
→ **greedy search is (at least) as good as graph-based assignment**
→ **the wellformedness constraint is important**
Results: cross-domain

What about overfitting?

Check if feature weights are stable across textual domains!
(Economy Texts in SMULTRON)

<table>
<thead>
<tr>
<th>setting</th>
<th>Precision</th>
<th>Recall</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>train&amp;test=novel</td>
<td>77.95</td>
<td>76.53</td>
<td>77.23</td>
</tr>
<tr>
<td>train&amp;test=economy</td>
<td>81.48</td>
<td>73.73</td>
<td>77.41</td>
</tr>
<tr>
<td>train=novel, test=economy</td>
<td>77.32</td>
<td>73.66</td>
<td>75.45</td>
</tr>
<tr>
<td>train=economy, test=novel</td>
<td>78.91</td>
<td>73.55</td>
<td>76.13</td>
</tr>
</tbody>
</table>

No big drop in performance! → Good!
Conclusions

- flexible classifier-based tree alignment framework
- rich feature set (+ context, + history)
- good results even with tiny amounts of training data
- relatively stable across textual domains
The End

Thanks!

Questions? Comments? Discussion?

http://stp.lingfil.uu.se/~joerg/trealigner
Compatible with Stockholm Tree Aligner
Alignment Features: Lexical Equivalence

\[ \gamma(s, t) = \alpha(s \mid t)\alpha(t \mid s)\alpha(s \mid \overline{t})\alpha(\overline{t} \mid s) \]

Our implementation of \( \alpha \)

\[ \alpha_{inside}(s \mid t) = \prod_{s_i \in \text{yield}(s)} \max_{t_j \in \text{yield}(t)} P(s_i \mid t_j) \]

\[ \alpha_{outside}(s \mid t) = \prod_{s_i \notin \text{yield}(s)} \max_{t_j \notin \text{yield}(t)} P(s_i \mid t_j) \]

GIZA++/Moses provide \( P(s_i \mid t_m) \) and \( P(t_m \mid s_i) \)
Alignment Features: Sub-tree Features

Features that describe the relative position differences of nodes within the trees:

- **tree-level similarity**: $1 - \text{difference in relative distance to root}$
- **tree-span similarity**: $1 - \text{difference in relative “horizontal” positions}$

Size difference:

- **leafratio**: ratio of terminal nodes dominated by current tree nodes
Subtree features

\[
\begin{align*}
    tls(s_i, t_j) &= 1 - \text{abs}\left(\frac{d(s_i, s_{root})}{\max_x d(s_x, s_{root})} - \frac{d(t_i, t_{root})}{\max_x d(t_x, t_{root})}\right) \\
tss(s_i, t_j) &= 1 - \text{abs}\left(\frac{s_{start} + s_{end}}{2 \times \text{length}(S)} - \frac{t_{start} + t_{end}}{2 \times \text{length}(T)}\right) \\
    \text{leafratio}(s_i, t_j) &= \frac{\min(|\text{leafnodes}(s_i)|, |\text{leafnodes}(t_j)|)}{\max(|\text{leafnodes}(s_i)|, |\text{leafnodes}(t_j)|)}
\end{align*}
\]
Well-formedness Constraint

“Descendants/ancestors of a source linked node may only be linked to descendants/ancestors of its target linked counterpart”

→ no inconsistent links
Results: compare node types

How good is the aligner on different node types?

<table>
<thead>
<tr>
<th>node type</th>
<th>Recall</th>
<th>Precision</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>non-terminals</td>
<td>78.08</td>
<td>82.32</td>
<td>80.15</td>
</tr>
<tr>
<td>terminals</td>
<td>71.79</td>
<td>78.00</td>
<td>74.77</td>
</tr>
</tbody>
</table>

Good on non-terminal nodes!
1:1 alignment constraints probably too strict for leaf nodes
Results: base features

How good are base features on their own?

<table>
<thead>
<tr>
<th>features</th>
<th>Prec</th>
<th>Rec</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>lexical</td>
<td>66.07</td>
<td>36.77</td>
<td>47.24</td>
</tr>
<tr>
<td>tree</td>
<td>30.46</td>
<td>34.50</td>
<td>32.36</td>
</tr>
<tr>
<td>alignment</td>
<td>61.36</td>
<td>54.52</td>
<td>57.74</td>
</tr>
<tr>
<td>label</td>
<td>36.14</td>
<td>35.12</td>
<td>35.62</td>
</tr>
<tr>
<td>context-label</td>
<td>56.53</td>
<td>44.64</td>
<td>49.88</td>
</tr>
</tbody>
</table>

Performance is low but promising!
(Very little training data and very simple features!)