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

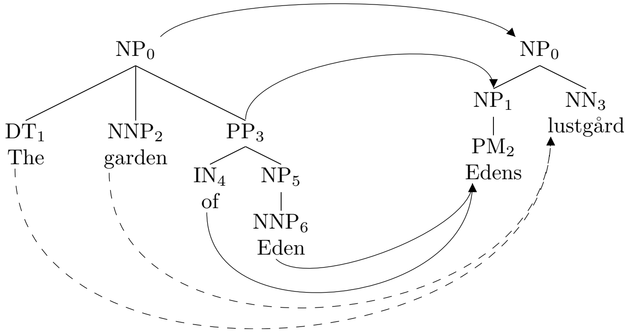
- phrase & rule extraction for (statistical) MT
- data for CAT, CALL applications
- corpus-based contrastive/translation studies

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

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

$\rightarrow$ learning task: find optimal feature weights λ_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 γ based on probabilistic bilingual lexicons ($P(s_l|t_m)$ and $P(t_m|s_l)$ created by GIZA++):

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

(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:* $\langle \bar{s}; \bar{t}\rangle$)

Alignment Features: Word Alignment

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

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

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

→ 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!

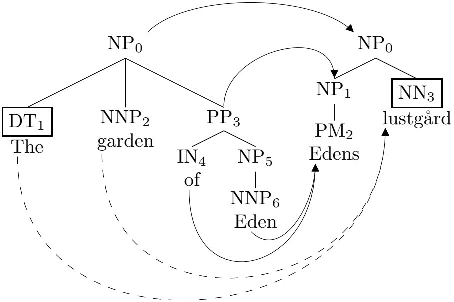
- local binary classifier: predictions in isolation
- implicit dependencies: include features from the context
- features of **parent nodes, child nodes, sister nodes, grandparents ...**

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

Example Features

Some possible features for node pair $\langle \mathrm{DT}_1, \mathrm{NN}_3 \rangle$

feature	value
labels=DT-NN	1
tree-span-similarity	0
tree-level-similarity	1
sister_labels=PP-NP	1
sister_labels=NNP-NP	1
parent_$\alpha_{inside}(t\|s)$	0.00001077
srcparent_GIZA$_{src2trg}$	0.75



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?)

$\rightarrow$ **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

$\rightarrow$ **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

$$Kuhn-Munkres\left(\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}\right) = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

$\rightarrow$ 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

$$Precision = \frac{|P \cap A|}{|A|} \qquad Recall = \frac{|S \cap A|}{|S|}$$

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$

S = sure ("good") links
P = possible ("fuzzy" + "good") links
A = links proposed by the system

Results on different feature sets (F-scores)

inference →	threshold=0.5		graph-assign	greedy	+wellformed
history →	no	yes			
lexical	38.52	40.00			
+ tree	50.27	51.84			
+ alignment	60.41	60.63			
+ labels	72.44	72.24			
+ context	74.68	74.90			

→ **additional features always help**

Results on different feature sets (F-scores)

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

→ **additional features always help**
→ **alignment inference is important (with weak features)**

Results on different feature sets (F-scores)

inference →	threshold=0.5		graph-assign		greedy		+wellformed
history →	no	yes	no	yes	no	yes	
lexical	38.52	40.00	49.75	56.60	50.05	56.76	
+ tree	50.27	51.84	54.41	57.01	54.55	57.81	
+ alignment	60.41	60.63	61.31	60.83	60.92	60.87	
+ labels	72.44	72.24	72.72	73.05	72.94	73.14	
+ context	74.68	74.90	74.96	75.38	75.03	75.60	

→ **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)

inference →	threshold=0.5		graph-assign		greedy		+wellformed	
history →	no	yes	no	yes	no	yes	no	yes
lexical	38.52	40.00	49.75	56.60	50.05	56.76	52.03	**57.11**
+ tree	50.27	51.84	54.41	57.01	54.55	57.81	57.54	**58.68**
+ alignment	60.41	60.63	61.31	60.83	60.92	60.87	62.09	**62.88**
+ labels	72.44	72.24	72.72	73.05	72.94	73.14	75.72	**75.79**
+ context	74.68	74.90	74.96	75.38	75.03	75.60	77.29	**77.66**

→ **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)

setting	Precision	Recall	F
train&test=novel	77.95	76.53	77.23
train&test=economy	81.48	73.73	77.41
train=novel, test=economy	77.32	73.66	75.45
train=economy, test=novel	78.91	73.55	76.13

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

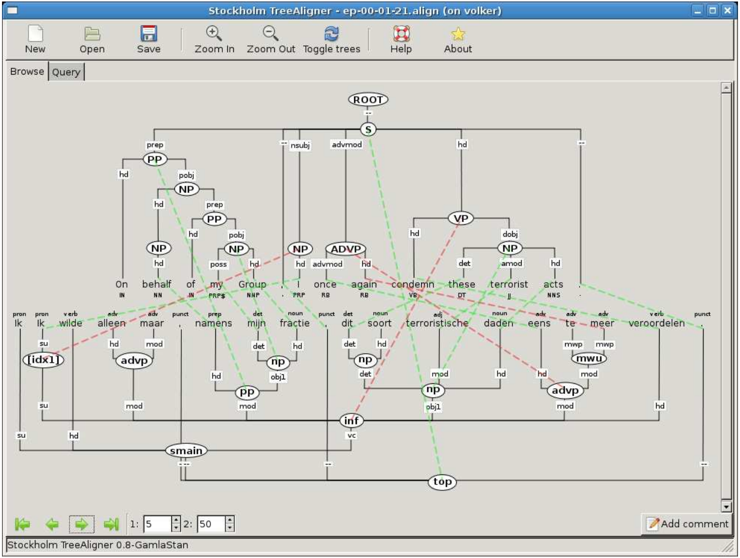
The End

Thanks!

Questions? Comments? Discussion?

http://stp.lingfil.uu.se/∼joerg/treealigner

Compatible with Stockholm Tree Aligner



UPPSALA UNIVERSITET

Alignment Features: Lexical Equivalence

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

Our implementation of α

$$\alpha_{inside}(s|t) = \prod_{s_i \in yield(s)} max_{t_j \in yield(t)} P(s_i|t_j)$$
$$\alpha_{outside}(s|t) = \prod_{s_i \notin yield(s)} max_{t_j \notin yield(t)} P(s_i|t_j)$$

GIZA++/Moses provide $P(s_l|t_m)$ and $P(t_m|s_l)$

Alignment Features: Sub-tree Features

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

tree-level similarity: 1 - difference in relative distance to root

tree-span similarity: 1- difference in relative “horizontal” positions

Size difference:

leafratio: ratio of terminal nodes dominated by current tree nodes

Subtree features

$$tls(s_i, t_j) = 1 - abs\left(\frac{d(s_i, s_{root})}{max_x d(s_x, s_{root})} - \frac{d(t_i, t_{root})}{max_x d(t_x, t_{root})}\right)$$

$$tss(s_i, t_j) = 1 - abs\left(\frac{s_{start} + s_{end}}{2 * length(S)} - \frac{t_{start} + t_{end}}{2 * length(T)}\right)$$

$$leafratio(s_i, t_j) = \frac{min(|leafnodes(s_i)|, |leafnodes(t_j)|)}{max(|leafnodes(s_i)|, |leafnodes(t_j)|)}$$

Well-formedness Constraint

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

$\rightarrow$ no inconsistent links

Results: compare node types

How good is the aligner on different node types?

node type	Recall	Precision	F
non-terminals	**78.08**	**82.32**	**80.15**
terminals	71.79	78.00	74.77

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?

features	Prec	Rec	F
lexical	**66.07**	36.77	47.24
tree	30.46	34.50	32.36
alignment	61.36	**54.52**	**57.74**
label	36.14	35.12	35.62
context-label	56.53	44.64	49.88

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