Detecting semantic similarity using syntactic tree alignment

Erwin Marsi, Emiel Krahmer

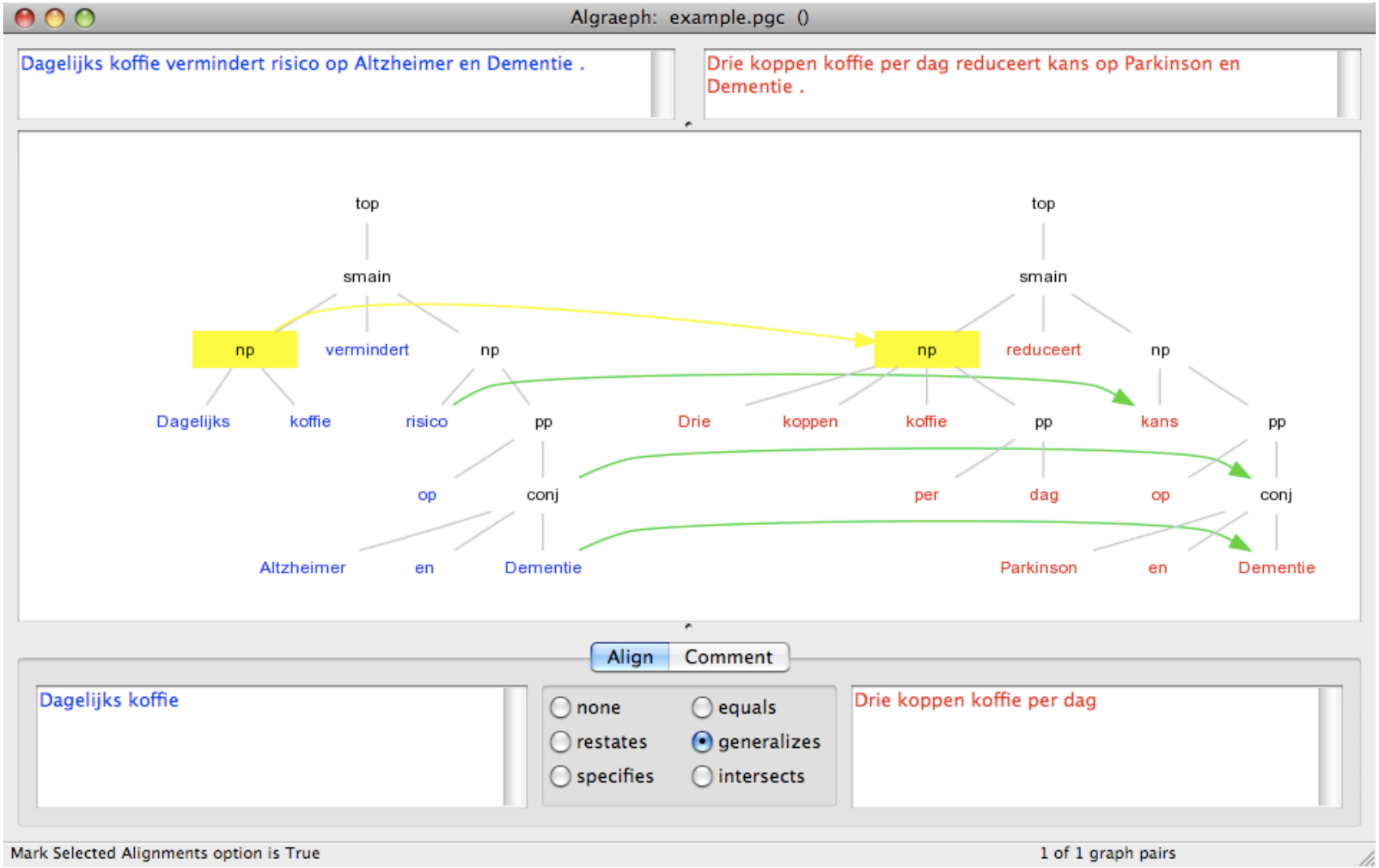
Universiteit van Tilburg

Friday February 5, 2010

Introduction

- **Goal**: automatically analyze to what extent two (Dutch) sentences have *similar meaning*
- **Motivation**: has many applications in e.g. multi-document summarization, QA, plagiarism detection, intelligent document merging,...
- **Approach**:
  - *align syntax trees*, where each node in the source tree may be aligned to the *most similar* node in the target tree
  - label alignments with a *semantic similarity relation*

Example


Algraeph: example.pgc ()
Dagelijks koffie vermindert risico op Altzheimer en Dementie .
Drie koppen koffie per dag reduceert kans op Parkinson en Dementie .
top
smain
np
vermindert
np
Dagelijks
koffie
risico
pp
op
conj
Altzheimer
en
Dementie
top
smain
np
reduceert
np
Drie
koppen
koffie
pp
kans
pp
per
dag
op
conj
Parkinson
en
Dementie
Align
Comment
Dagelijks koffie
none
equals
restates
generalizes
specifies
intersects
Drie koppen koffie per dag
Mark Selected Alignments option is True
1 of 1 graph pairs

Semantic similarity relations

Source *Dagelijks koffie vermindert risico op Altzheimer en Dementie.*

Target *Drie koppen koffie per dag reduceert kans op Parkinson en Dementie.*

- *Dementie* **equals** *Dementie*
- *vermindert* **restates** *reduceert*
- *dagelijks koffie* **generalizes** *drie koppen koffie per dag*
- *drie koppen koffie per dag* **specializes** *dagelijks koffie*
- *Altzheimer en Dementie* **intersects** *Parkinson en Dementie*

Outline

1. Daeso corpus
2. Tree alignment algorithm
3. Word alignment experiments
4. Tree alignment experiments
5. Conclusion

Daeso Corpus

- a monolingual (Dutch) treebank of parallel/comparable text
- aligned at the level of sentences and syntactic nodes (includes word alignment)
- node alignments are labeled acording to 5 similarity relations
- DAESO = *Detecting And Exploiting Semantic Overlap*

Text material

1. Books
   - Darwin - Origin of Species
   - Montaigne - Essays
   - Saint-Exupery - Le Petit Prince
2. Autocue-subtitle
   - pairs from NOS and VRT Journals
3. Headlines
   - headlines from similar news articles from Google News
4. News
   - sentences from press releases by ANP and Novum about the same news event
5. QA
   - alternative answers to the same question in a QA context
   - obtained from IMIX QA reference corpus

Corpus construction

1. tokenize texts with DCOI tokenizer
2. sentence alignment (partly manual)
3. parsing with Alpino parser
4. tree alignment and relation labeling (mostly manual)

Alignment counts

Source	#Graphpairs	#Nodepairs	#Tokens
Autocue-subtitle	9 851	135 798	217 956
Books	3 430	63 874	114 485
Headlines	13 084	89 086	97 681
News	8 248	86 227	162 361
QA	186	1 503	2 230
Totals:	34 799	376 488	594 713

Distribution of relations per corpus segment

Segment	Eq	Re	Spec	Gen	Inter
Autocue-subtitle	67.46	11.48	2.58	14.12	4.37
Books	57.17	21.87	3.82	4.31	12.84
Headlines	54.56	11.03	9.48	10.43	14.49
News	55.59	8.32	7.58	7.05	21.46
QA	59.28	6.05	5.59	4.79	24.28
Overall:	58.89	12.13	6.33	10.02	12.64

Automatic alignment of syntax trees

- Tree alignment & labeling is a hard problem
  - alignment & labeling are closely related tasks
  - knowledge from different types/sources comes into play
  - corpus data is noisy
- We have developed a new model for graph/tree alignment
- Casts alignment and labeling tasks simultaneously as a combination of
  1. generic *classification* with a machine learner
  2. global optimization of alignments using a *combinatorial optimization* algorithm

Pipeline for automatic tree alignment

Input: pair of Alpino parse trees (in CGN XML format)

1. feature extraction
2. relation classification
3. weighting
4. matching

Output: node matching (list of source- and target node pairs)

Step 1: Feature extraction

- For each possible pairing of a source node $sn_1, \ldots sn_n$ and a target node $tn_1, \ldots tn_m$, create an instance (i.e. $n * m$ instances) where
  - the features express a property of individual nodes or node pairs
  - the class label is any of the five semantic similarity relations or none (i.e. unaligned)

Step 1: Feature extraction (cont'd)

- We can extract all sorts of shallow and deep features
  - same (lower-cased) word/pharse
  - substring/prefix/suffix/infix
  - same roots
  - same POS / category
  - lenght difference
  - parent nodes have same category
  - overlap in dependency triples
  - lexical semantic relations (e.g. synonym, hyponym, hyperonym) from Cornetto
  - matching digits to numerical expressions
  - matching abbreviation (*e.g*) to exanded strings (*for example*)
  - ...

Step 2: Relation classification

- Given the instances for each possible pairing of source to a target node, use a generic machine learner to predict the alignment relation (majority class is *none*)
- Currently we are using Timbl (IB1)
- Results in *classification clashes*
  - a source node is aligned to multiple target nodes, and vice versa
  - we see a node *mapping* (n-to-n), but we want a node *matching* (1-to-1)...

Step 3: Weighting

- assign a *cost* (akin to a confidence score) to each *alignment* + *relation* prediction
- currently we use the *normalized entropy* of the class labels in the set of nearest neighbours (but there are other options)
  - if all NN are of the same class, then $entropy = 0.0$
  - if NN are of different classes, then $0 < entropy <= 1.0$

Step 4: Matching

- given $n * m$ possible alignments and associated costs, we want to find
  - a node *matching*, i.e. only one-to-one alignments
  - which *minimizes the sum of costs* over all alignments
- this is a well-known problem in combinatorial optimization known as the *Assignment Problem*
- in graph-theoretical terms: find a *minimum weighted bipartite graph matching*
- can be solved in polynomial time ($O(n^3)$) using e.g. the *Hungarian algorithm* (Kuhn, 1955)

News data set

- similar press releases from ANP and Novum
- comparable text
- manually aligned and labeled for relation
- 86420 node alignments
- 40571 of which are word (i.e. terminal-node) alignments

Features on Words

No	Name
1	source-word-uniq
2	target-word-uniq
3	same-words-lhs
4	same-words-rhs
5	word-len-diff
6	words-subsumption
7	words-shared-prefix-len
8	words-shared-infix-len
9	words-shared-suffix-len

Features on Roots

No	Name
10	roots-subsumption
11	roots-share-prefix
12	roots-share-infix
13	roots-share-suffix

Features on POS

No	Name
14	source-pos
15	target-pos
16	same-pos

Features using Cornetto

No	Name
17	cornet-restates
18	cornet-specifies
19	cornet-generalizes
20	cornet-intersects

Features on Syntax

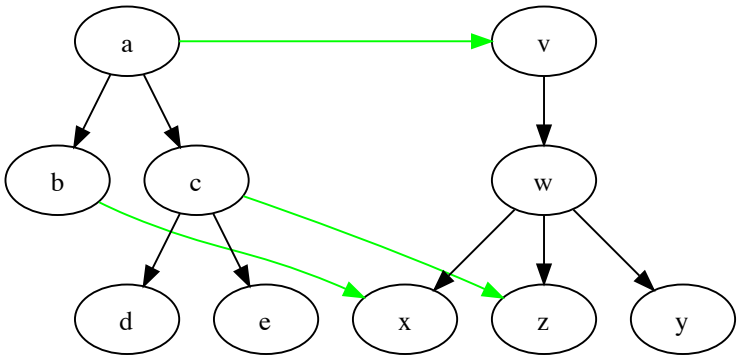
No	Name
21	source-cat
22	target-cat
23	same-cat
24	source-parent-cat
25	target-parent-cat
26	same-parent-cat
27	source-dep-rel
28	target-dep-rel
29	same-dep-rel
30	same-dep-head-root

Classifier & Weight function

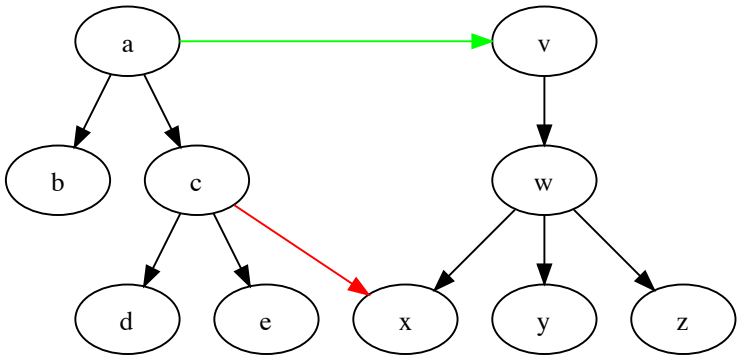
- classifier: Timbl with default setting
- weighting: normalized entropy in nearest neighbour set

Precision, recall and F-score on alignment

True alignment



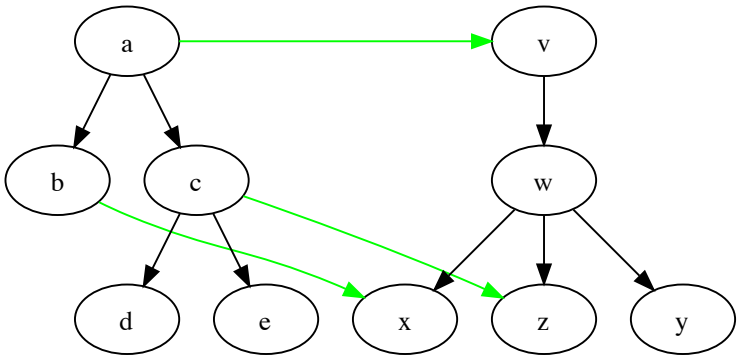
Predicted alignment



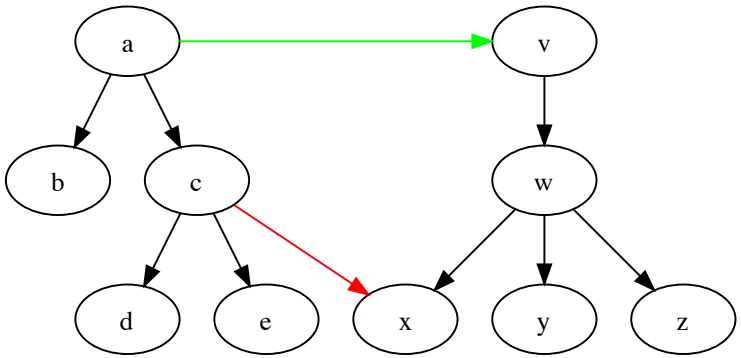
- $Precision = \frac{|True \cap Pred|}{|Pred|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<c,x>\}|} = \frac{1}{2} = 0.5$

Precision, recall and F-score on alignment

True alignment



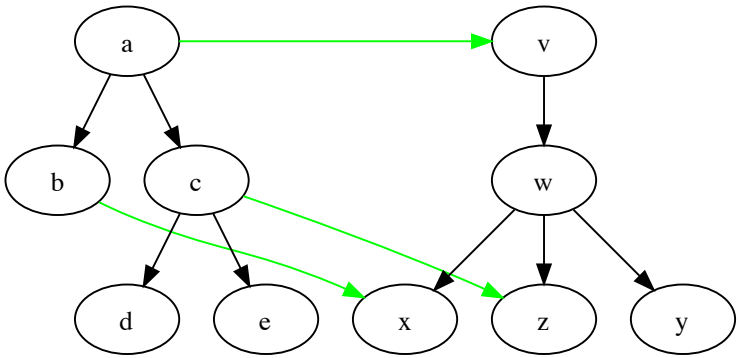
Predicted alignment



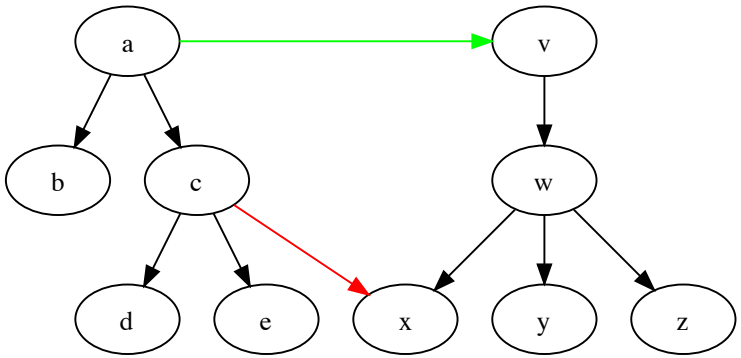
- $Precision = \frac{|True \cap Pred|}{|Pred|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<c,x>\}|} = \frac{1}{2} = 0.5$
- $Recall = \frac{|True \cap Pred|}{|True|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<b,x>,<c,z>\}|} = \frac{1}{3} = 0.33$

Precision, recall and F-score on alignment

True alignment



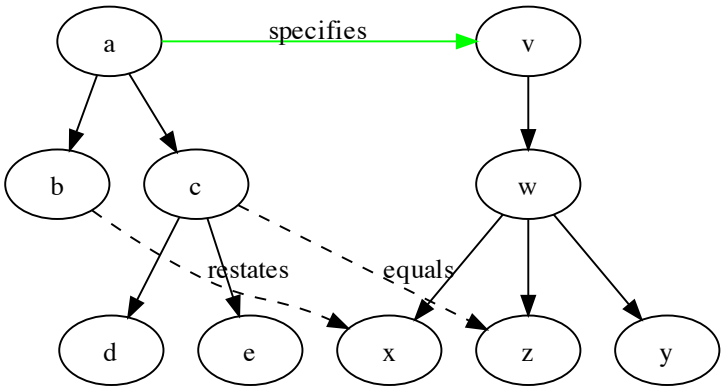
Predicted alignment



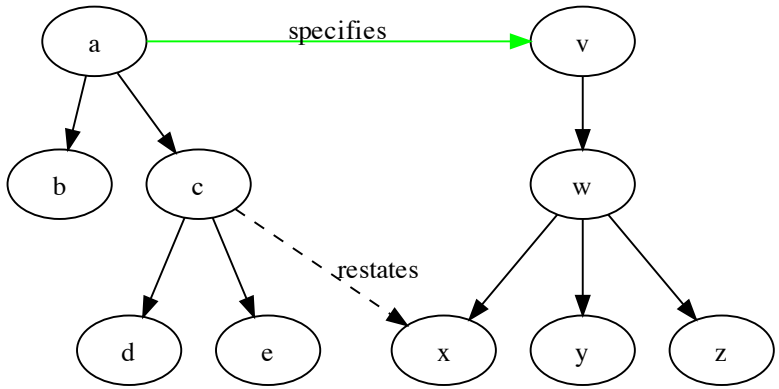
- $Precision = \frac{|True \cap Pred|}{|Pred|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<c,x>\}|} = \frac{1}{2} = 0.5$
- $Recall = \frac{|True \cap Pred|}{|True|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<b,x>,<c,z>\}|} = \frac{1}{3} = 0.33$
- $F_1 score = \frac{2 * precision * recall}{precision + recall} = \frac{2 * 1/2 * 1/3}{1/2 + 1/3} = \frac{2}{5} = 0.4$

Precision, recall and F-score on relation labeling

True alignment



Predicted alignment



- To calculate scores on relation R:
  1. restrict to alignments labeled with relation R
  2. calculate precision, recall and F-score as before
- Repeat for each relation R

Baseline

- Baseline: greedy alignment of equal words
  - align identical source and target words (in order) as *equals*
  - align identical source and target roots (in order) as *restates*
  - dumb but quite effective because *equals* is by far the majority class
  - always scores zero on the *specifies*, *generalizes* and *intersects* relations

Baseline scores on News data

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	81.84	93.10	87.11
restates	46.26	34.71	39.66
specifies	0.00	0.00	0.00
generalizes	0.00	0.00	0.00
intersects	0.00	0.00	0.00
Macro Mean:	25.62	25.56	25.35
Micro Mean:	80.22	81.20	80.70

Daeso Aligner on News data (10-CV)

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	95.92	94.51	95.21
restates	60.90	42.20	49.85
specifies	61.68	28.70	39.17
generalizes	70.55	42.12	52.75
intersects	41.27	23.64	30.06
Macro Mean:	66.06	46.23	53.41
Micro Mean:	91.96	85.45	88.58

Upper bound

- We measured inter-annotator agreement
  - on small samples of corpus
  - with 6 annotators
  - using a jack knife procedure
    - take one annotator as reference, and calculate prec, rec and F-score for the other 5 annotators
    - repeat for each annotator
    - take average over all $6 * 5$ scores

Average Human on sample of News data

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	95.38	95.38	95.38
restates	58.50	58.50	58.50
specifies	65.81	65.81	65.81
generalizes	65.00	65.00	65.00
intersects	25.85	25.85	25.85
Macro Mean:	62.11	62.11	62.11
Micro Mean:	88.72	88.72	88.72

Daeso Aligner on News data - relative to Average Human

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	+0.54	-0.87	-0.17
restates	+2.40	-16.30	-8.65
specifies	-4.13	-37.11	-26.64
generalizes	+5.55	-22.88	-12.25
intersects	+15.42	-2.21	+4.21
Macro Mean:	+3.95	-15.88	-8.7
Micro Mean:	+3.24	-3.27	-0.14

Experiments on full tree alignment

- same data set of comparable news text
- same feature set plus some additional features
- downsampling to reduce the number of instances of class *none*
- taking more nearest neighbors into account (k=15)

Extra features in tree alignment

No	Name
31	token-prec
32	token-rec
33	same-lc-phrase
34	same-words-rhs
35	source-phrase-len
36	target-phrase-len
37	phrase-len-diff

Baseline scores on News data

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	86.96	93.67	90.19
restates	0.00	0.00	0.00
specifies	0.00	0.00	0.00
generalizes	0.00	0.00	0.00
intersects	0.00	0.00	0.00
Macro Mean:	17.39	18.73	18.04
Micro Mean:	86.96	51.94	65.04

Daeso Aligner on News data (10-CV)

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	94.03	96.83	95.41
restates	43.47	37.42	40.22
specifies	52.97	37.69	44.04
generalizes	53.71	41.41	46.77
intersects	59.29	67.11	62.96
Macro Mean:	60.69	56.09	57.88
Micro Mean:	77.97	77.46	77.72

Average Human on sample of News data

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	95.83	95.83	95.83
restates	71.38	71.38	71.38
specifies	60.21	60.21	60.21
generalizes	66.71	66.71	66.71
intersects	62.67	62.67	62.67
Macro Mean:	71.36	71.36	71.36
Micro Mean:	81.92	81.92	81.92

Daeso Aligner on News data - relative to Average Human

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	-1.80	+1.00	-0.42
restates	-27.91	-33.96	-31.16
specifies	-7.24	-22.52	-16.17
generalizes	-13.00	-25.03	-19.94
intersects	-3.46	+4.44	+0.29
Macro Mean:	-10.67	-15.27	-13.48
Micro Mean:	-3.95	-4.46	-4.2

Conclusion

- We presented a new model for tree/graph alignment which simultaneously performs alignment and relation labeling using
  1. generic *classification* with a machine learner
  2. global optimization of alignments using a *combinatorial optimization* algorithm
- Preliminary results on word alignment show reasonable results
  - at least consistently above the baseline
  - but still substantially below the average human score on the non-equal relations
  - phrase alignment is harder than word alignment

Future work

- Use predicted word alignments as input for full tree alignment. For example:
  - if the words in two phrases are aligned, then the phrases are more likely to be aligned
  - if their words are aligned as restates & equals, then the phrases are likely to be aligned as restates
- Compare to other approaches (word aligners used in SMT)
- Extension/tuning/optimization