An algorithm for detecting semantic similarity using syntactic tree alignment

Erwin Marsi, Emiel Krahmer

Universiteit van Tilburg

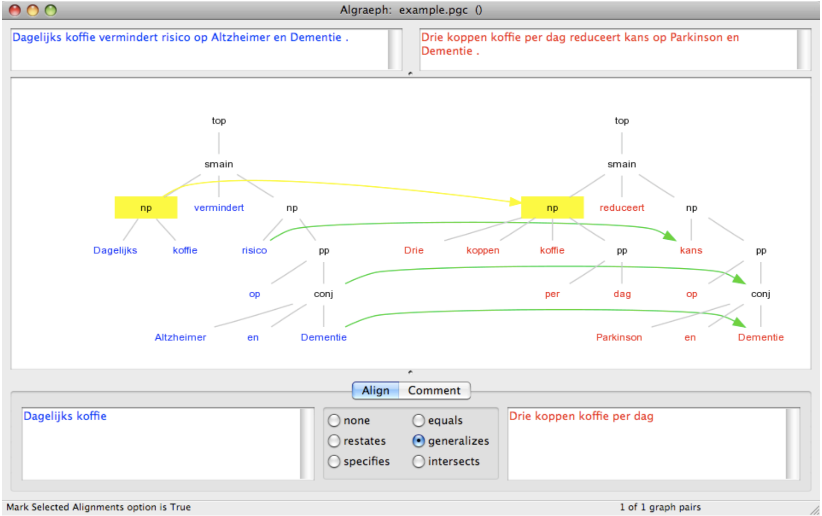
Tuesday November 10, 2009

Outline

Introduction

- Goal: 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 their syntactic trees, where each node in the source tree may be aligned to another node in the target tree

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

Daeso corpus

- Daeso corpus
  - a monolingual (Dutch) treebank of parallel/comparable text
  - aligned at the level of sentences and nodes
  - node alignments labeled acording to five similarity relations

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 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

On the task of 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
- Our initial approach using DP-based tree alignment algorithms did not work very well
  - restrictions on matching are too strict (no crossing alignments)
  - difficult to combine knowledge sources in a non-arbitrary way
  - linguistic structures may not be trees
- Hence matching needs to be more relaxed (error-tolerant graph matching)
- Further attempts suffered from
  - lack of integration between alignment and relation labeling
  - need for manual tuning of arbitrary weights/thresholds
  - knowledge-based hacks per relation/domain

Automatic alignment of syntax trees (cont'd)

- Recently 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 (null,nill)

Step 1: Feature extraction (cont'd)

- We can extract all sorts of shallow and deep features
  - string identity
  - substring
  - Levenshtein distance
  - same root
  - POS / category
  - parent node has 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 (usually *none*)
- Currently we are using Timbl (IB1)
- Notice that this often results in “classification clashes”
  - a source node may become aligned to multiple target nodes, and vice versa
  - we get a node *mapping*, but we want a node *matching*. . .

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 the *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)

Related work

- There is a huge body of work on text alignment
- Similarly for global optimization methods in ML for NLP
- Alignment in terms of the Assignment Problem
  - Taskar, Lacoste-Julien and Klein, *A Discriminative Matching Approach to Word Alignment*, EMNLP 05
  - ...
- Other NLP tasks using the Assignment Problem
  - Wan, Dras, Dale, Paris, *Improving Grammaticality in Statistical Sentence Generation*, ACL 09
  - ...

Cheat sheet

- I'm cheating: experimental results about word alignment, not full tree alignment
- Excuses:
  - easier to explain (fewer features)
  - could not get the full experiments up and running in time
  - aligning non-terminals seems to require taking previous decisions (i.e. alignments in the subtree) into account

Data sets

- Books (parallel text)
  - two alternative translations of *Le Petit Prince*
  - 8229 node alignments
- News (comparable text)
  - similar press releases from ANP and Novum
  - 13027 node alignments

Feature extraction

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

Feature extraction (cont'd)

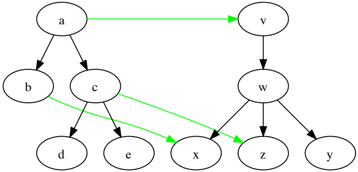
No	Name
10	roots-subsumption
11	roots-share-prefix
12	roots-share-infix
13	roots-share-suffix
14	source-pos
15	target-pos
16	same-pos
17	cornet-restates
18	cornet-specifies
19	cornet-generalizes
20	cornet-intersects

Classifier & Weight function

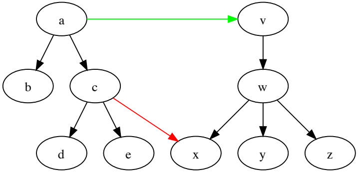
- classifier: Timbl with default setting
- weighting: normalized entropy in nearest neighbour set

Precision, recall and F-score on alignment

True alignment



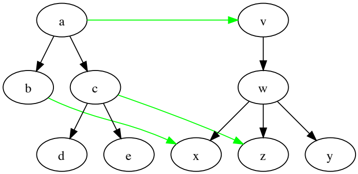
Predicted alignment



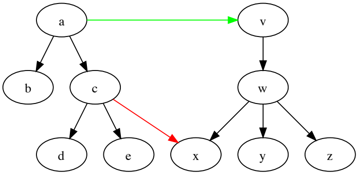
- $Precision = \frac{|True \cap Pred|}{|Pred|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<c,x>\}|} = \frac{1}{2} = 0.5$

Precision, recall and F-score on alignment

True alignment



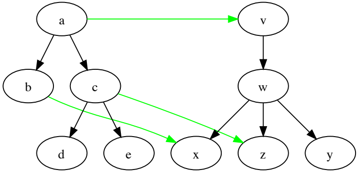
Predicted alignment



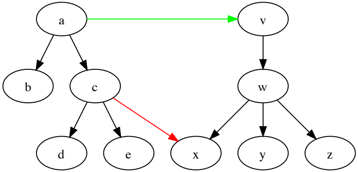
- $$Precision = \frac{|True \cap Pred|}{|Pred|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<c,x>\}|} = \frac{1}{2} = 0.5$$
- $$Recall = \frac{|True \cap Pred|}{|True|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<b,x>,<c,z>\}|} = \frac{1}{3} = 0.33$$

Precision, recall and F-score on alignment

True alignment



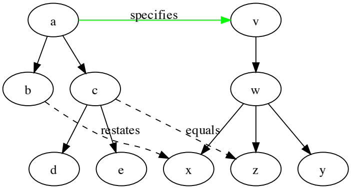
Predicted alignment



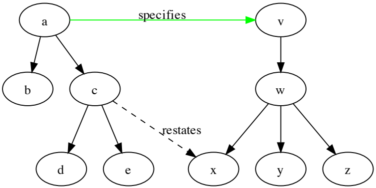
- $Precision = \frac{|True \cap Pred|}{|Pred|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<c,x>\}|} = \frac{1}{2} = 0.5$
- $Recall = \frac{|True \cap Pred|}{|True|} = \frac{|\{<a,v>\}|}{|\{<a,v>,<b,x>,<c,z>\}|} = \frac{1}{3} = 0.33$
- $F_1 score = \frac{2 * precision * recall}{precision + recall} = \frac{2*1/2*1/3}{1/2+1/3}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

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)
  - dumb but quite effective because *equals* is by far the majority class
  - always scores zero on non-equal relations

Upper bound

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

Baseline scores on Books.Saint data

Relation	Prec (%)	Rec (%)	F (%)
equals	93.39	95.55	94.46
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:	18.68	19.11	18.89
Micro Mean:	93.39	81.09	86.81

Daeso Aligner on Books.Saint data (10-CV)

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	97.81	94.22	95.98
restates	69.01	34.13	45.67
specifies	35.71	12.20	18.18
generalizes	42.86	19.48	26.79
intersects	22.64	9.84	13.71
Macro Mean:	53.61	33.97	40.07
Micro Mean:	94.94	84.52	89.42

Daeso Aligner on Books.Saint data - relative to baseline

Relation	Prec (%)	Rec (%)	F (%)
equals	+4.42	-1.33	+1.52
restates	+69.01	+34.13	+45.67
specifies	+35.71	+12.20	+18.18
generalizes	+42.86	+19.48	+26.79
intersects	+22.64	+9.84	+13.71
Macro Mean:	+34.93	+14.86	+21.18
Micro Mean:	+1.55	+7.43	+2.61

Average Human on sample of Books.Saint data

Relation	Prec (%)	Rec (%)	F (%)
equals	95.86	95.86	95.86
restates	73.89	73.89	73.89
specifies	60.00	60.00	60.00
generalizes	59.61	59.61	59.61
intersects	23.90	23.90	23.90
Macro Mean:	62.65	62.65	62.65
Micro Mean:	88.95	88.95	88.95

Daeso Aligner on Books.Saint - relative to Average Human

Relation	**Prec** (%)	**Rec** (%)	**F** (%)
equals	+1.95	-1.64	+0.12
restates	-4.88	-39.76	-28.22
specifies	-24.99	-47.80	-41.82
generalizes	-16.75	-40.13	-32.28
intersects	-1.26	-14.06	-10.19
Macro Mean:	-9.04	-28.68	-22.58
Micro Mean:	+5.99	-4.43	+0.47

Baseline scores on News data

Relation	Prec (%)	Rec (%)	F (%)
equals	81.85	93.17	87.14
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:	16.37	18.63	17.43
Micro Mean:	81.85	79.12	80.46

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

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

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 only show promising results
    - at least consistently above the baseline
    - but still substantially below the average human score on the non-equal relations
    - precision is better than the recall

Future work

- Run experiments on full tree alignment
- Compare to other approaches
  - word aligners used in SMT
- Extension/tuning/optimization
  - feature selection: many other types of features possible
  - classifier settings
  - different alignment weighting measures
  - forced alignment & threshold on the max alignment cost