http://www.stanford.edu/~icard/logic&language/

http://www.stanford.edu/~icard/logic&language/djalali-potts-natlog.pdf

http://modestconsequences.wordpress.com/

NatLog implementation

General Overview

Pipeline:

- Linguistic analysis
- Alignment
- Lexical Entailment classification
- Entailment projection
- Entailment joining

Linguistic analysis

As explained before except that the NatLog system as implemented uses the PS parse of the Stanford system, not the dependency graph.

Monotonicity marking of tokens in the input span (e.g. ***without*** is downward monotone). For each relevant operator the arity is specified and a tree pattern that helps identify its occurrence in the Treebank parse.DOWN and NON are marked, upward monotonicity is considered to be the default. Note that the other entailment projection signatures are ignored in the implementation.

No scope disambiguation; assumes that scope depends on linear order

Alignment

Remember

- EQ: connects a span in p with an equal span in h (based on lemmas)
- SUB: connects two unequal spans
- DEL: covers an unaligned span in p
- INS: covers an unaligned span in h.

The monotonicity marking is used in the heuristic rules that order the edits: in general DEL before SUB before INS; but downward monotone and non-monotone operators are grouped after SUB and before INS.

Lexical entailment classification

Based on machine learning exploiting external lexical resources about the relationships between word meanings

Feature representation

A real-valued vector encoding information about the edit: type, size, characteristics of the words, and (for SUB) information about the semantic relation between substituends.

Features for DEL and INS:

Light: boolean, + when semantically light word (punctuation, preposition, possessive, article, auxiliary, expletive)
Pronoun: boolean: + when pronoun
MiscDel: hand-coded mappings from specific expressions: DEL(not) yields ^, DEL(true) yields ≡; non-intersective adjectives: DEL(fake) or DEL(former) yields |, DEL(alleged) yields #; mappings for implicatives and factives: DEL (force) yields ⊏, DEL(fail) yields ^. For the factives, the positive ones (e.g. to know) are given a lexical ⊏ signature and the negative ones (e.g. to pretend) a | one.

Features for SUB

WordNet-derived features

WNSyn (synonymy) 1 iff the substituends are synonymous, 0 otherwise
WNAnt (antonymy): 1 iff the substituends are antonyms, 0 otherwise
WNHyper (hypernymy): if the h phrase is a hypernym of the p phrase, WNHyper takes the value 1- n/8, where n is the number of links to get from p to h, otherwise the value is 0; ex. WNHyper (owl,bird) = 0.75, WNHyper(bird,owl) = 0,0.
WNHypo (Hyponymy) inverse of WNHyper
JiCo: semantic relatedness based on distance between two WN synsets, the measure used in the one proposed in Jiang and Conrath 1997.

Features based on other lexical resources

NomB: NomBank has related noun/verb and adjective/adverb pairs. Nomb is set to 0.75 if the substituends are among these pairs, 0 otherwise.

DLin: semantic relatedness based on a thesaurus complied by Dekang Lin, based on distributional similarity, mapping depends on the score given in the thesaurus

String similarity features

LemStrSim similarity based on the string edit distance between word lemmas

$$\mathrm{LemStrSim}(w1,w2)=\max\left[0,1-\frac{dist(lemma(w1),lemma(w2))}{\max(|lemma(w1)|,|lemma(w2)|)-k}\right]$$

where the distance is the Levenshtein string edit distance and k is a penalty parameter for very short string. It is put to 2, so the LemStrSim between ‘she’ and ‘the’ is 0 but the LemStrSim between ‘Admadinejad’ and ‘Ahmadinejab’ is 0.89

LemSubSeq identifies the case where one of the substituends is a multi-word phrase which contains the other as a sub-phrase

LemSubSeq(p,h) = ⊏ iff h is a subsequence of p

LemSubSeq(p,h) = ⊐ iff p is a subsequence of h

LemSubSeq(p,h) = ≡ iff h and p are equal

LemSubSeq(p,h) = # otherwise

Lexical category features

Light: boolean: *on* when both substituends are semantically light.

Preps: boolean: *on* when both substituends are prepositions -- NatLog is liberal with preps.

Pronoun:boolean: *on* when both substituends are pronouns -- NatLog is liberal with pronouns.

NNN: boolean: *on* when both substituends are either common nouns or proper nouns, this predisposes the relation to being |

Quantifier: defines about a dozen quantifier categories and the relations e.g. SUB(ALL,SOME) → ⊏

Miscellaneous features

NeqNum: boolean: *on* if both substituends are numbers and they are not equal

MiscSub: handcoded mappings for specific pairs of expressions, e.g. SUB(and,or) → ⊏; after, before → |, etc.

Classifier

Decision tree (J48)

- Training set 2,449 lexical entailment problems: 1,525 SUB edits (words or phrases), 924 DEL edits (one word) , manually annotated with one of the seven relations

examples:
SUB(blast,explosion): ≡
SUB(acquired, bought): ⊐
Most SUBs were ≡, followed by |
Most DELs were ⊏, followed by ≡
99% accuracy on the training data.

Entailment projection

Based on the monotonicity marking done in the parsing stage.

Then: take the smallest parse constituent containing the tokens involved in the edit, trace a path upward through the parse tree from there to the root, collect monitonicity markers along the way. If any of these is NON, conclude NON for the whole, otherwise, if the number of DOWN markers is odd, conclude DOWN, if it is even, conclude UP.

If we end up with NON we conclude #, if we end up with DOWN we assume that every lexical entailment is projected as its dual under negation, if we end up with UP, we assume everything is projected without change

Dual under negation: $\forall x,y : <x,y> \in R \Leftrightarrow <\overline{x},\overline{y}> \in S$; R1011 and R1101 (bit strings are reversed), R1001is its own dual.

Result: $\sqsubset$ and $\sqsupset$ are inverted.

Entailment joining

As described before.

Tendency to get # (conservative in its outcomes)

```
(S
  (NP (NNP Jimmy) (NNP Dean))
  (VP (VBD refused)
       (S
          (VP (TO to)
              (VP (VB move)
                  PP (IN without)
                    (NP (NNS jeans)))))))
```

```
(S
  (NP (NNP James) (NNP Dean))
  (VP (VBD did) (RB n't)
        (VP (VB dance)
                  PP (IN without)
                    (NP (NNS pants)))))))
```

edit	feature	lexical rel	monotonicity	projection	join
SUB(Jimmy Dean, James Dean)	strsim:0.67	≡	UP	≡	≡
SUB(move,dance)	hyponym	⊐	DOWN	⊏	⊏
EQ (without,without)		≡	DOWN	≡	⊏
SUB(jeans, pants)	hypernym	⊏	UP	⊏	⊏
DEL(refused to)	implic:+-	|	UP	|	|
INS(n't)	cat: neg	^	UP	^	⊏
INS(did)	cat: aux	≡	DOWN	≡	⊏

Evaluation: Fracas

Test suite created to evaluate inferential phenomena. 346 problems

Asks for YES, NO, UNKNOWN answers

59% problems have YES answers, 28% UNKNOWN and 10% NO

154 contain multiple premises, excluded from the evaluation. 183 left

Example of multiple premise reasoning:

Smith wrote a report in two hours,

Smith started writing the report at 8 a.m.,

--> Smith had finished writing the report by 11 a.m.

Fracas

	P	R	Acc
baseline: assume yes	55.7	100.0	55.7
bag of words	59.7	87.2	57.4
NatLog 8	89.3	65.7	70.5

accuracy: correct/total
precision: correct(tp)/correct+fp
recall: correct/correct+fn

Fracas

guess gold	yes	no	unknown	total
yes	67	4	31	102
no	1	16	4	21
unknown	7	7	46	60
total	75	27	81	183

Fracas contains problems sets that NatLog doesn't know anything about: ellipsis, anaphora resolution, temporal reference, NatLog does well on quantifiers (97.7), adjectives (80.0), attitudes (88.9)

Evaluation:RTE3

	P	R	Acc
Stanford	68.8	60.2	60.5
Hybrid	64.5	68.9	64.5

Evaluation:RTE3

	P	R	Acc
Stanford	68.8	60.2	60.5
Hybrid	64.5	68.9	64.5

best	80.0
average	62.4
median	62.6

Evaluation:RTE3

	P	R	Acc
Stanford	68.8	60.2	60.5
Hybrid	64.5	68.9	64.5

best	80.0
average	62.4
median	62.6

bag of words	62.8