Natural Logic for Textual Inference

B. MacCartney & C. Manning 2007

Presentation by Christian Wellner

Overview

- Introduction
- Natural Logic
- The NatLog System
- Experiments with FraCaS
- Experiments with RTE
- Conclusion

Introduction

- Area of interest: Textual inference
  - Infer a *hypothesis* from a *premise*
- Recent (2005-2006) approaches:
  - Measures of lexical or semantic overlap
  - Pattern-based relation extraction
  - Matching of predicate-argument structure
  - Translation to first-order logic (FOL)

Introduction

- Problems with Non-FOL-systems:
  - Very robust, but imprecise
  - Confused by (negative) monotonicity:

    P: No case of indigenously acquired rabies infection has been confirmed in the past 2 years.

    H: No rabies cases have been confirmed.

    → H does not follow, but lexical content and predicate-argument structure can be matched

Introduction

- Problems with FOL-systems:
  - High precision at the cost of low recall (e.g. Bos and Markert 2006)
  - Translation of natural language to FOL is tricky, e.g. for idioms, intensionality, modalities, temporal relations, quantifiers, ...

Introduction

- This paper's approach: Natural Logic
  - Not to be confused with Natural deduction, which is a proof system for FOL
  - Based on natural language, not logical notation
  - Theoretical background: Monotonicity calculus from Sánchez Valencia 1991

Introduction

- This paper's approach: Natural Logic
  - Cannot „solve“ some types of inference (e.g. temporal reasoning, relation extraction)
  - But avoids the problems of translating sentences into FOL while preserving precision
  - Overall broad applicability, especially if combined with other techniques

Natural Logic

- Problem to solve:
  Does a premise P entail a hypothesis H?
  - Try to incrementally transform P into H
  - Use reasoning about monotonicity

Natural Logic

- Semantic constraints can (sometimes) be expanded or contracted salva veritate

  Ex.: „Every meal without wine is a terrible crime“

  - Positive polarity / Expansion: terrible crime → crime
  - Negative polarity / Contraction: meal → dinner

  => entailment relation

Natural Logic

- Entailment relation „$\sqsubseteq$“
  - For c,d $\in$ <t>: c $\sqsubseteq$ d iff c $\rightarrow$ d
  - For c,d $\in$ <e>: c $\sqsubseteq$ d iff c = d
  - For c,d $\in$ <A,B>: c $\sqsubseteq$ d iff
    for all a $\in$ A: c(a) $\sqsubseteq$ d(a)
  - c # d iff c $\not\sqsubseteq$ d and d $\not\sqsubseteq$ c

Natural Logic

- Entailment relation „⊑“
  - Nouns: penguin ⊑ bird
  - Adjectives: French ⊑ European
  - Verbs: kick ⊑ strike
  - Temporal modifiers: this morning ⊑ today
  - Local modifiers: in Beijing ⊑ in China
  - Connectives: and ⊑ or

Natural Logic

- Entailment relation „⊑“
  - In general, dropping a modifier yields entailment: eat quickly ⊑ eat
  - But not always: fake vaccine ⋢ vaccine
  - Also controversial: quantifiers (e.g. everyone ⊑ someone, which is different from FOL)

Natural Logic

- Monotonicity:
  - A function f is upwards-monotone iff
    for all $x \leq y$: $f(x) \leq f(y)$
  - f is downwards-monotone iff
    for all $x \leq y$: $f(y) \leq f(x)$
  - f is non-monotone iff it is neither upward- nor downward-monotone

Natural Logic

- Monotonicity:
  - A property of functions → the meaning of a compound expression is calculated by function application!
    - e.g. Exp1<a→b> (Exp2<a>) → Exp1Exp2<b>
  - Instead of numeric order („≤“), use entailment order („⊑“)

Natural Logic

- Monotonicity:
  - Most semantic functions are upward-monotone: tango in Paris $\sqsubseteq$ dance in France, since tango $\sqsubseteq$ dance and Paris $\sqsubseteq$ France
  - However, some very important constructions are downward-monotone, like negation: tango $\sqsubseteq$ dance, but didn't dance $\sqsubseteq$ didn't tango

Natural Logic

- Monotonicity:
  - Some words can be either depending on context: every is downward-monotone in every fish swims $\sqsubseteq$ every shark swims, but upward-monotone in every shark swims $\sqsubseteq$ every shark moves
  - Natural logic, unlike many other approaches, allows to distinguish between these cases

Natural Logic

- Monotonicity in composition:
  - Suppose h = f o g
  - If either f or g is non-monotone, so is h
  - If monotonicities of f and g are the same, h is upward-monotone/ has positive polarity
  - If monotonicities of f and g are different, h is downward-monotone / has negative polarity

Natural Logic

- Monotonicity in composition:
  - $f(x) = 2x$; $g(x) = 2x \rightarrow h(x) = 4x$
  - $f(x) = 2x$; $g(x) = -2x \rightarrow h(x) = -4x$
  - $f(x) = -2x$; $g(x) = 2x \rightarrow h(x) = -4x$
  - $f(x) = -2x$; $g(x) = -2x \rightarrow h(x) = 4x$

The NatLog System

- The NatLog System consists of 3 parts:
  - 1) Linguistic preprocessing
  - 2) Alignment
  - 3) Entailment classification

The NatLog System

- Linguistic preprocessing
  - Use Statistical parser for tokenization, POS tagging and phrase structure parsing
  - Use monotonicity calculus to mark each token span with its monotonicity
  - Some adaptions because phrase structure is not always the same as semantic structure (manual patterns to match spans)

The NatLog System

- Alignment
  - Transformation of Premise into Hypothesis, done by a sequence of 4 atomic edits:
    - Deletion of a token span from H
    - Insertion of a token span into H
    - Substitution of a token span of H
    - Advance without edit if token spans match

The NatLog System

- Alignment
  - Example:

An Irishman	⟹	An Irishman	ADV
won	⟹	won	ADV
a	⟹	the	SUB
Nobel prize	⟹	Nobel prize	ADV
	⟹	for literature	INS
.	⟹	.	ADV

The NatLog System

- Alignment
  - No movement edit possible (only indirectly by DEL + INS)
  - Levenshtein algorithms can be used to compute alignment efficiently
  - All intermediate steps between P and H are well-defined

The NatLog System

- Alignment
  - Heuristic edit cost function for Levenshtein computation:
    - Edits on longer spans are preferred
    - Edits on non-constituent-spans are penalized
    - Little cost for „light“ edits like punctuation, articles, prepositions etc.
    - Not deeply optimized

The NatLog System

- Entailment classification
  - Global entailment problem is now split into a sequence of atomic entailment problems
  - Predict an entailment relation for each atomic edit and compose these predictions to get the global prediction

The NatLog System

- Entailment classification
  - Elementary entailment relations:

relation	symbol	in terms of $\sqsubseteq$	FraCaS	RTE
equivalent	$p = h$	$p \sqsubseteq h, h \sqsubseteq p$	yes	yes
forward	$p \sqsubset h$	$p \sqsubseteq h, h \not\sqsubseteq p$	yes	yes
reverse	$p \sqsupset h$	$h \sqsubseteq p, p \not\sqsubseteq h$	unk	no
independent	$p \# h$	$p \not\sqsubseteq h, h \not\sqsubseteq p$	unk	no
exclusive	$p \mid h$	$p \sqsubseteq \neg h$	no	no

Table 1: The five elementary entailment relations. The last two columns indicate correspondences to FraCaS and RTE answers; see sections 4 and 5.

The NatLog System

- Entailment classification
  - Atomic classifier: Decision tree that uses as features:
    - Type of edit (ADV, DEL, INS, SUB)
    - Importance of edit (is it „light“ or not)
    - Monotonicity of the token span
    - „various lexical features“ that indicate synonymy, hyponymy, antonymy etc.
    - Lemma similarity based on Levenshtein

The NatLog System

- Entailment classification
  - Example: An Irishman won a nobel prize → An Irishman won the nobel prize for literature

atomic edit: SUB(*a*, *the*)
  features:
    type: SUB, *monotonicity*: ↑, *isLightEdit*: true,
    wnSyno: 0.0, *wnHypo*: 0.0, *wnAnto*: 0.0, *lemmaSim*: 0.0
  predicted entailment relation: =

atomic edit: INS(*for literature*)
  features:
    type: INS, *monotonicity*: ↑, *isLightEdit*: false
  predicted entailment relation: ⊐

top-level inference:
  composition of entailment relations: = ∘ ⊐ ⇒ ⊐
  mapping to FraCaS answer: ⊐ ⇒ *unk*

Experiments with FraCaS

- FraCaS test suite (Cooper et al. 1996)
  - consists of 346 inference problems
  - 9 different categories of problems
  - Each problem consists of one ore many premises and one question
  - Possible answers: Yes (H can be inferred), No (negation of H can be inferred), Unk (neither)

Experiments with FraCaS

- Application of NatLog
  - All FraCaS problems with more than one premise or no clear hypothesis were excluded
  - Questions were converted to declarations
  - NatLog is expected to have expertise only in some of the 9 categories (namely 1, 5 and 6)

Experiments with FraCaS

- Application of NatLog
  - Examples:

§	ID	Premise(s)	Hypothesis	Ans
1	33	An Irishman won a Nobel prize.	An Irishman won the Nobel prize for literature.	*unk*
1	38	No delegate finished the report.	Some delegate finished the report on time.	*no*
2	99	Clients at the demonstration were all impressed by the system's performance. Smith was a client at the demonstration.	Smith was impressed by the system's performance.	*yes*
9	335	Smith believed that ITEL had won the contract in 1992.	ITEL won the contract in 1992.	*unk*

Experiments with FraCaS

- Results:
  - Accuracy:

§	Category	Count	% Acc.
1	Quantifiers	44	84.09
2	Plurals	24	41.67
3	Anaphora	6	50.00
4	Ellipsis	25	28.00
5	Adjectives	15	60.00
6	Comparatives	16	68.75
7	Temporal	36	61.11
8	Verbs	8	62.50
9	Attitudes	9	55.56
Applicable sections: 1, 5, 6		75	76.00
All sections		183	59.56

Experiments with FraCaS

- Results:
  - As expected, the 3 „applicable“ categories yield better results
  - Some others do as well
  - Ellipsis problems as to be expected are solved with lowest accuracy

Experiments with FraCaS

- Results:
  - Confusion Matrix:

	guess			
answer	*yes*	*unk*	*no*	total
yes	62	40	–	102
unk	15	45	–	60
no	6	13	2	21
total	90	91	2	183

Experiments with FraCaS

- Results:
  - FraCaS is biased towards Yes
  - NatLog wasn't trained on FraCaS data
  - NatLog tends towards Unk: If one atomic prediction „fails“, the prediction for the whole sequence cannot be determined anymore
  - NatLog sometimes incorrectly answers Yes: probably due to cases like fake vaccine

Experiments with RTE

- RTE Challenge (Dagan et al. 2005)
  - More natural examples than FraCaS
  - Longer premises (average 35 words vs. 11)
  - Only binary classification: No and Unk merged

Experiments with RTE

- RTE Challenge (Dagan et al. 2005)
  - Example:

ID	Premise(s)	Hypothesis	Answer
518	The French railway company SNCF is cooperating in the project.	The French railway company is called SNCF.	*yes*
601	NUCOR has pioneered a giant mini-mill in which steel is poured into continuous casting machines.	Nucor has pioneered the first mini-mill.	*no*

Table 5: Illustrative examples from the RTE3 test suite

Experiments with RTE

- RTE Challenge (Dagan et al. 2005)
  - Mostly not really suited for NatLog
  - NatLog could however improve performance on a subset of RTE problems if combined with another system
  - In this paper: Stanford RTE system of Maneffe et al. 2006, its alignment is used as input for NatLog step 3

Experiments with RTE

- Hybridization:
  - Stanford system makes predictions by using a treshold score, which is adjusted by +x or -x depending on NatLog choice
  - Balanced Hybrid: Threshold is optimized first, then x is trained to balance Yes and No
  - Optimized Hybrid: x is fixed after finding balance, threshold is then optimized again

Experiments with RTE

- Results:

RTE3 Development Set (800 problems)				
System	% yes	precision	recall	accuracy
Stanford	50.25	68.66	66.99	67.25
NatLog	18.00	**76.39**	26.70	58.00
Hybrid, bal.	50.00	69.75	67.72	68.25
Hybrid, opt.	55.13	69.16	**74.03**	**69.63**

RTE3 Test Set (800 problems)				
System	% yes	precision	recall	accuracy
Stanford	50.00	61.75	60.24	60.50
NatLog	23.88	**68.06**	31.71	57.38
Hybrid, bal.	50.00	64.50	62.93	63.25
Hybrid, opt.	54.13	63.74	**67.32**	**63.62**

Experiments with RTE

- Results:
  - The hybrid systems outperform both isolated systems
  - Interestingly, RTE data does not contain a lot of inferences about monotonicity
  - NatLog seems to generally improve precision

Conclusion

- NatLog applicability:
  - Limited on several kinds of inferences
  - However, it can be used to support other systems by providing higher precision
  - There are some possiblities for optimization of NatLog itself

Conclusion

- Future Work:
  - Optimization of alignment cost function (hybrid probably had better alignment)
  - In some steps, manual definitions are used which certainly could be improved
  - NatLog only uses one premise, could be extended to multiple premises

Thank you for your attention