# Abduction for Deep Linguistic Reasoning and RTE

HS Linguistic Inference and
Textual Entailment
Prof. Dr. Manfred Pinkal

29.05.2007
Anna Mündelein

# Outline

- Introduction
  - What Is Abduction?
- Abductive Inference
  - TACITUS System
  - Interpretation as Abduction
  - Weighted Abduction
  - Solving Pragmatic Problems
  - Combining Syntax, Semantics and Pragmatics
- An Abductive Approach to RTE
  - Motivation
  - The Approach in Detail
  - Results
- Conclusion

# What Is Abduction?

- imagine you are looking out the window and see a tree waving back and forth – what is your explanation of the tree's movement?
- possible explanations:
  - "The wind is blowing."
  - "There is a man standing below window level and shaking the tree."
  - ...
- most people would go for the first alternative because it is the most plausible one
  - most economical: only few and "normal" assumptions necessary
  - consistent with what we know
- explanations of this kind: **abductive**

# What Is Abduction?

- "**Abduction**, or **inference to the best explanation**, is a method of reasoning in which one chooses the hypothesis which would, if true, best explain the relevant evidence." [Wikipedia]

| | **Deduction** | **Induction** | **Abduction** |
|---|---|---|---|
| Premises | $\forall x(p(x) \rightarrow q(x))$ $p(A)$ | several instances $p(A), q(A)$ | $\forall x(p(x) \rightarrow q(x))$ $q(A)$ |
| Conclusion | $q(A)$ | $\forall x(p(x) \rightarrow q(x))$ | $p(A)$ |

- only deduction is valid
- but: abduction is the only logical operation that introduces new ideas

# TACITUS System

- **T**he **A**bductive **C**ommonsense **I**nference **T**ext **U**nderstanding **S**ystem
  - processing of messages and other texts for a variety of purposes
    - e.g. equipment failure reports – perform diagnosis
  - aim: investigate how knowledge is used in the interpretation of discourse
  - large knowledge base of commonsense and domain knowledge

# Interpretation as Abduction

- to interpret a sentence:
  - present its content as predications (logical form)
  - prove the predications by using the axioms in the knowledge base
  - allow assumptions in your proof, at various costs
  - pick the proof with the lowest cost

# Interpretation: Example

*The Boston office called.*

- three pragmatic problems:
  - reference resolution: *the Boston office*
  - metonymy resolution: an office cannot call; what is meant is *Some person at the Boston office called*
  - compound nominal interpretation: determine the implicit relation between *Boston* and *office*
- logical form:

$$\exists x \exists y \exists z \exists e \left( call(e, x) \wedge person(x) \wedge office(y) \wedge rel(x, y) \wedge Boston(z) \wedge nn(z, y) \right)$$

# Interpretation: Example

- assume the knowledge base contains:

$$Boston(B_1)$$
$$office(O_1) \wedge in(O_1, B_1)$$
$$person(J_1)$$
$$work-for(J_1, O_1)$$
$$\forall y \forall z (in(y, z) \rightarrow nn(z, y))$$
$$\forall x \forall y (work-for(x, y) \rightarrow rel(x, y))$$

# Interpretation: Example

$$\boxed{call'(e,x)} \wedge person(x) \wedge rel(x,y) \wedge office(y) \wedge Boston(z) \wedge nn(z,y)$$

**Knowledge Base:**

$person(J_1)$

$work\text{-}for(x,y) \supset rel(x,y)$

$work\text{-}for(J_1,O_1)$

$office(O_1)$

$Boston(B_1)$

$in(y,z) \supset nn(z,y)$

$in(O_1,B_1)$

# Weighted Abduction

- when parts of the expression cannot be derived, assumptions have to be made
- assumptions: new information
- likelihood for the different conjuncts to be new information varies

  → assign cost to each conjunct

  – example:
    - definite noun phrase: \$10
    - indefinite noun phrase: \$1
    - verb: \$3

$$\ldots \wedge office(y)^{\$10} \wedge call(e,x)^{\$3} \wedge \ldots$$

# Weighted Abduction

- scheme of abductive inference:
  1) every conjunct in the logical form of a sentence is given an assumability cost:
  $$... \wedge Q^c \wedge R^d \wedge \dots$$
  2) this cost is passed back from the consequent literal to the antecedent literals in implications:
  $$P_1^{\omega_1} \wedge P_2^{\omega_2} \rightarrow Q$$
     - the cost of assuming $P_1$ is $\omega_1 c$ and the cost of assuming $P_2$ is $\omega_2 c$
     - if $\omega_1 + \omega_2 < 1$ → most-specific abduction
     - if $\omega_1 + \omega_2 > 1$ → least-specific abduction

# Weighted Abduction

3) factoring allowed:

$$\exists x \exists y \exists \ldots \left( \ldots \wedge q(x)^{\$20} \wedge \ldots \wedge q(y)^{\$10} \wedge \ldots \right)$$
$$\rightarrow \exists x \exists \ldots \left( \ldots \wedge q(x)^{\$10} \wedge \ldots \right)$$

→ exploits natural redundancy of texts

e.g. *Inspection of oil filter revealed metal particles.*

- the weights can be chosen according to how much each conjunct contributes semantically to the implication

$$\forall x \left( car(x)^{0.8} \wedge no-top(x)^{0.4} \rightarrow convertible(x) \right)$$

# “Et Cetera” Propositions

- usually axioms like:

$$\forall x(elephant(x) \rightarrow mammal(x))$$

- problem: in the abductive approach backward-chaining and not forward-chaining is used
  - if we encounter $elephant(x)$ in the text, we cannot do anything with it
- solution: $\forall x(mammal(x)^{0.2} \wedge etc(x)^{0.9} \rightarrow elephant(x))$
  - the *etc* predicate can never be proven, but we can assume it

# Solving pragmatic problems

- the abductive inference approach provides solutions to several pragmatic problems, e.g.:
  - distinguishing the given and the new information in a sentence
  - lexical ambiguity
  - compound nominal interpretation

# Solving pragmatic problems

- distinguishing the given and the new information in a sentence
  - example:

    *John walked into the room.*

    *The chandelier shone brightly.*

  - what chandelier is being referred to?
  - if we simply assume a chandelier, it cannot be linked to the room

  → knowledge base:

$$\forall r\big(room(r) \rightarrow \exists l\big(light(l) \wedge in(l, r)\big)\big)$$

$$\forall l\big(light(l) \wedge has-branches(l) \rightarrow chandelier(l)\big)$$

# Solving pragmatic problems

**Logical Form:**

$$\ldots \wedge \mathit{chandelier}(x) \wedge \ldots$$

**Knowledge Base:**

$$\mathit{light}(l) \wedge \boxed{\mathit{has\text{-}branches}(l)} \supset \mathit{chandelier}(l)$$

$$\mathit{room}(r) \supset \mathit{light}(l) \wedge \mathit{in}(l, r)$$

$$\mathit{room}(R)$$

# Solving pragmatic problems

- lexical ambiguity
  - example:

    *John wanted a loan. He went to the bank.*

  → knowledge base:

$$\forall x\left(bank_1(x) \rightarrow bank(x)\right)$$

$$\forall x\left(bank_2(x) \rightarrow bank(x)\right)$$

$$\forall y\left(loan(y) \rightarrow \exists x\left(financial-institution(x) \wedge issue(x, y)\right)\right)$$

$$\forall x\left(financial-institution(x) \wedge etc_1(x) \rightarrow bank_1(x)\right)$$

$$\forall z\left(river(z) \rightarrow \exists x\left(bank_2(x) \wedge borders(x, z)\right)\right)$$

# Solving pragmatic problems

**Logical Form:**

$$\ldots \wedge bank(x) \wedge \ldots$$

**Knowledge Base:**

$$bank_1(x) \supset bank(x)$$

$$financial\text{-}institution(x) \wedge \boxed{etc_1(x)} \supset bank_1(x)$$

$$loan(y) \supset financial\text{-}institution(x) \wedge issue(x,y)$$

$$loan(L)$$

$$bank_2(x) \supset bank(x)$$

$$river(z) \supset bank_2(x) \wedge borders(x,z)$$

# Solving pragmatic problems

- compound nominals:
  - examples:

    *Boston office* $\ldots \wedge Boston(x) \wedge office(y) \wedge nn(x,y) \wedge \ldots$

    *wine bottle* $\ldots \wedge wine(x) \wedge bottle(y) \wedge nn(x,y) \wedge \ldots$

    *oil sample* $\ldots \wedge oil(x) \wedge sample(y,z) \wedge nn(x,y) \wedge \ldots$
  - different types of relations between the nouns
  - → express all relations as *nn*, but write different axioms for *nn*:

$$\forall x \forall y (in(y,x) \rightarrow nn(x,y))$$

$$\forall e \forall x \forall y (contain(e,y,x) \rightarrow nn(x,y))$$

$$\ldots$$

# Combining Syntax, Semantics and Pragmatics

- combining the ideas of:
  - interpretation as abduction
  - parsing as deduction
- example (again):

  *The Boston office called.*

  - grammar:

$$\forall w_1 \forall w_2 (np(w_1) \wedge verb(w_2) \rightarrow s(w_1\ w_2))$$

$$\forall w_1 \forall w_2 (det(the) \wedge noun(w_1) \wedge noun(w_2) \rightarrow np(the\ \ w_1\ w_2))$$

  - to parse a sentence *W* is to prove *s(W)*

# Combining Syntax, Semantics and Pragmatics

- augment the axioms of the grammar with portions of the logical form:

$$\forall w_1 \forall w_2 \forall y \forall p \forall e \forall x \big( np(w_1, y) \wedge verb(w_2, p) \wedge p(e, x) \wedge rel(x, y) \wedge Req(p, x) \rightarrow s(w_1\ w_1, e) \big)$$

$$\forall w_1 \forall w_2 \forall q \forall r \forall y \forall z \big( det(the) \wedge noun(w_1, r) \wedge noun(w_2, q) \wedge r(z) \wedge q(y) \wedge nn(z, y) \rightarrow np(the\ w_1\ w_2, y) \big)$$

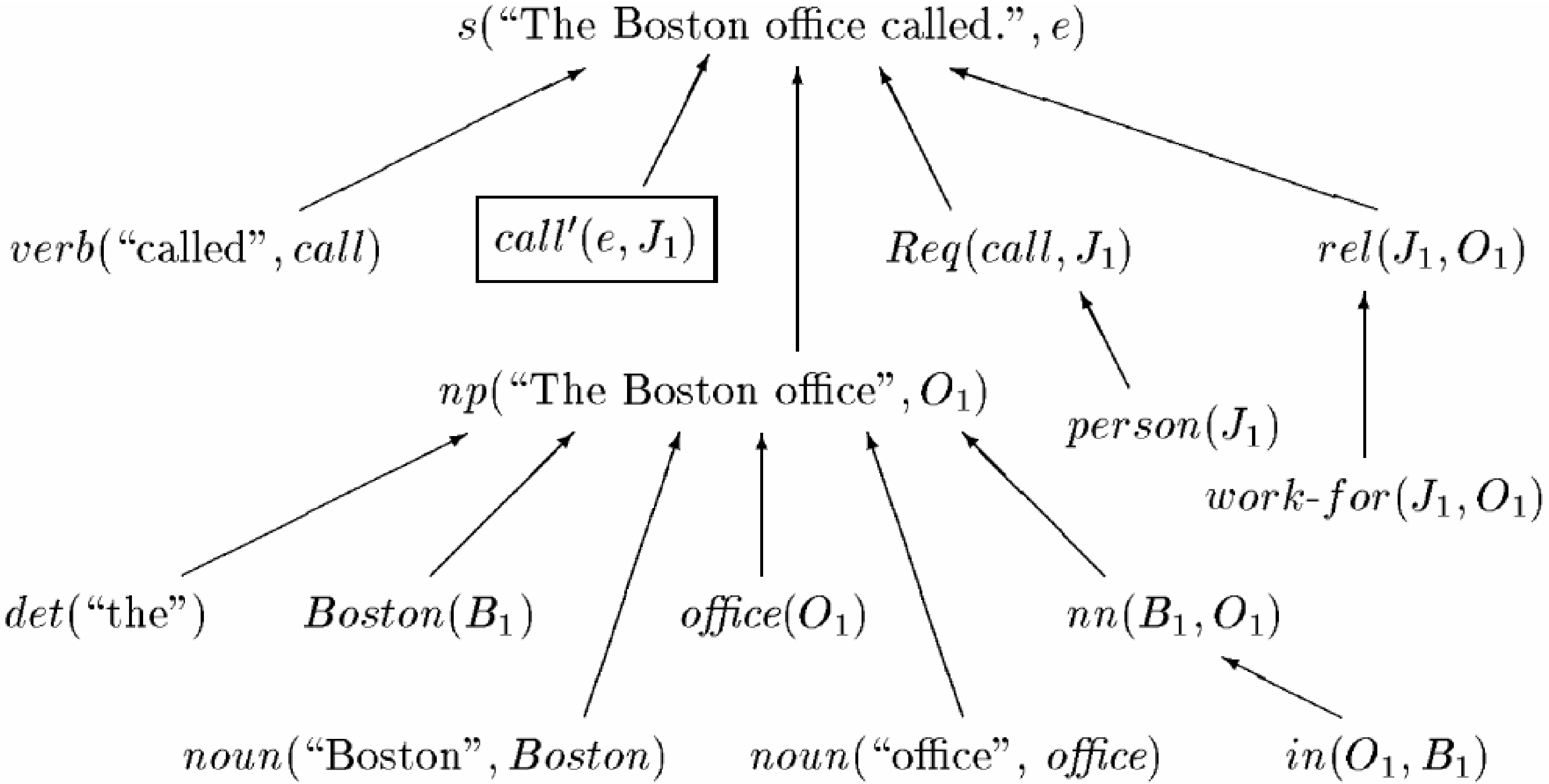
$p, r, q$ : correspond to *call, Boston, office*

$verb(w_2, p)$ : the string $w_2$ is a verb referring to $p$

$Req(p, x)$ : requirements that $p$ places on $x$

$$\forall x \big( person(x) \rightarrow Req(call, x) \big)$$

# Combining Syntax, Semantics and Pragmatics



# So far

- definition of abduction
- how abduction can be used to interpret (prove) texts
- but what about real applications?

→ use abduction for recognizing textual entailment

# An Abductive Approach to RTE

[Raina et al. 2005]

- Idea: combine deep linguistic reasoning and machine learning
  - use abductive reasoning to decide whether a text entails a hypothesis or not
    - logical formalism
    - elegant and precise
  - learn automatically which assumptions are plausible
    - statistical methods
    - robust and scalable

# Motivation

- given: text – hypothesis pair

  e.g. *TEXT: Bob purchased an old convertible.*

  *HYPOTHESIS: Bob bought an old car.*

- in bag-of-words representations there would be no difference between the hypothesis and a hypothesis like

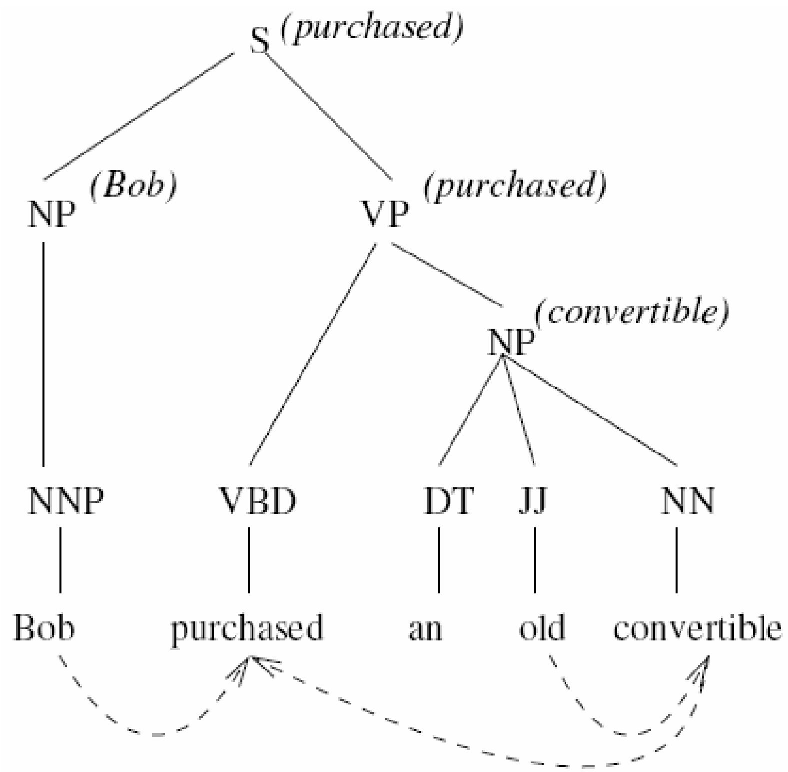
  *Old Bob bought a car.*

→ deeper representation needed

# The Approach in Detail

1) parse text and hypothesis
   - hand-written rules to find the heads of all nodes in the parse tree
   - head discovery leads to a kind of **dependency graph**

   TEXT:



# The Approach in Detail

2) transform the relations from the two dependency graphs into logical formulas
   - convert each node into a logical term with a unique constant (e.g. $NP^{(Bob)} \rightarrow Bob(A)$ )
   - represent edges by sharing arguments across nodes

TEXT:

$$\exists A\ \exists B\ \exists C\ \left(Bob(A) \wedge convertible(B) \wedge old(B) \wedge purchased(C,A,B)\right)$$

HYPOTHESIS:

$$\exists X\ \exists Y\ \exists Z\ \left(Bob(X) \wedge car(Y) \wedge old(Y) \wedge bought(Z,X,Y)\right)$$

# The Approach in Detail

3) augment the logical formulas with semantic annotations
   - annotations added on predicates (e.g. if the corresponding word is part of a named entity)
     - “*Bob* is a person”
   - annotations added on arguments (e.g. if the argument has a subject/object relation to its predicate)
     - “*convertible* is the object to *purchased*”

# The Approach in Detail

4) find a proof for the hypothesis given the text
   - resolution refutation proof:
     - add the axioms from the text to the knowledge base
     - add the negation of the hypothesis to the knowledge base
     - derive the null clause through successive resolution steps (unification of terms)
       → contradiction → hypothesis entailed in text

# The Approach in Detail

- knowledge base:
  - axiom from TEXT:

$$\exists A\,\exists B\,\exists C\;\left(Bob(A)\wedge convertible(B)\wedge old(B)\wedge purchased(C,A,B)\right)$$

  - negation of HYPOTHESIS:

$$\neg\left(\exists X\,\exists Y\,\exists Z\;\left(Bob(X)\wedge car(Y)\wedge old(Y)\wedge bought(Z,X,Y)\right)\right)$$

$$\leftrightarrow \forall X\,\forall Y\,\forall Z\;\left(\neg Bob(X)\vee\neg car(Y)\vee\neg old(Y)\vee\neg bought(Z,X,Y)\right)$$

- unification:
  - $Bob(A), \neg Bob(X)$
  - $old(B), \neg old(Y)$
  - but what about $convertible(B), \neg car(Y)$ and $purchased(C,A,B), \neg bought(Z,X,Y)$ ?

# The Approach in Detail

- unification of $S(s_1, s_2, \ldots, s_m)$ and $\neg T(t_1, t_2, \ldots, t_n)$
  - standard definition:
    - $S = T$
    - $m = n$
    - each $s_i$ consistently unified with $t_i$
  - relaxed definition:
    - $S$ and $T$ might be different
    - the numbers of arguments $m$ and $n$ need not be the same
    - two constant arguments could unify with each other

# The Approach in Detail

- relaxations: abductive assumptions about the world
- assign a cost to each relaxation depending on its degree of plausibility

→ **assumption cost model:**

the cost $C_\omega$ of assumption $A$ is

$$C_\omega(A) = \sum_{d=1}^{D} \omega_d f_d(A)$$

$f_1, \ldots, f_D$ : arbitrary nonnegative feature functions

$\omega_1, \ldots, \omega_D$ : relative weights assigned to the feature functions

# The Approach in Detail

- features can be derived from different knowledge sources (e.g. *WordNet*)
- five feature classes:

1) predicate similarity
   - synonyms / similar meaning
   - antonyms (if one predicate is negated)
2) predicate compatibility
   - same POS
   - same word stem
   - same named entity tag (if any)

# The Approach in Detail

3) argument compatibility
   - e.g. prefer subject argument to be matched with another subject argument
4) constant unification
   - different constants might refer to the same physical entity (e.g. because of anaphoric coreference)
   - compute "distance" between constants
5) word frequency
   - very commonly used terms can be ignored at some cost (e.g. *rather*)

# The Approach in Detail

1. Similarity score for $S$ and $T$.
   Are $S$ and $T$ antonyms?
   If $S$ and $T$ are numeric, are they "compatible"?
2. Mismatch type for part-of-speech tags of $S$ and $T$.
   Do $S$ and $T$ have same word stem?
   Do $S$ and $T$ have same named entity tag?
3. Difference in number of arguments: $|m - n|$.
   Number of matched arguments with:
   - different dependency types.
   - different semantic roles.
   - each type of part-of-speech mismatch.

   Number of unmatched arguments of $T$.
4. Total coreference "distance" between matched constants.
5. Inverse word frequency of predicate $S$.
   Is $S$ a noun, pronoun, verb or adjective, and is it being "ignored" by this unification?

# The Approach in Detail

- proof $P = A_1, A_2, \ldots, A_N$
- aggregated feature functions for the proof:

$$f_d(P) = \sum_{s=1}^{N} f_d(A_s)$$

- total cost of the proof:

$$C_\omega(P) = \sum_{d=1}^{D} \omega_d f_d(P) = \omega^T f(P)$$

- consider all possible proofs and pick the one with minimal cost
- if the minimal cost is below a certain threshold → classify hypothesis as entailed
- weights of the vector $\omega^T$ are chosen automatically by a learning algorithm

# Results

- participated in the PASCAL Recognizing Textual Entailment Challenge 2005

| Algorithm | RTE Dev Set | | RTE Test Set | |
|---|---|---|---|---|
| | Acc | CWS | Acc | CWS |
| Random | 50.0% | 0.500 | 50.0% | 0.500 |
| TF | 52.1% | 0.537 | 49.5% | 0.548 |
| TFIDF | 53.1% | 0.548 | 51.8% | 0.560 |
| ThmProver1 | **57.8%** | 0.661 | 55.5% | 0.638 |
| ThmProver2 | 56.1% | **0.672** | **57.0%** | **0.651** |
| Partial1 | 53.9% | 0.535 | 52.9% | 0.559 |
| Partial2 | 52.6% | 0.614 | 53.7% | 0.606 |

ThmProver1 : single threshold for all RTE classes

ThmProver2 : separate threshold for each RTE class

Partial1 : allows only standard unification

Partial2 : allows unification only when predicates match exactly

# Results

| Class | RTE Dev Set | | RTE Test Set | |
| --- | --- | --- | --- | --- |
| | Acc | CWS | Acc | CWS |
| CD | 71.4% | 0.872 | 79.3% | 0.906 |
| IE | 50.0% | 0.613 | 49.2% | 0.577 |
| IR | 50.0% | 0.523 | 50.0% | 0.559 |
| MT | 53.7% | 0.708 | 58.3% | 0.608 |
| PP | 62.2% | 0.685 | 46.0% | 0.453 |
| QA | 54.4% | 0.617 | 50.0% | 0.485 |
| RC | 47.6% | 0.510 | 53.6% | 0.567 |

- performance varies heavily by class
- accuracy of 57% competitive with the best reported result (58.6%)
- CWS of 0.651 significantly higher than for all other systems (next best result 0.617)

# Conclusion

- abduction is an interesting method for linguistic reasoning
- based on the way humans make inferences
- can be used for a variety of purposes
  - good results for recognizing textual entailment

# References

[Hobbs et al. 1988] Jerry R. Hobbs, Mark Stickel, Paul Martin and Douglas Edwards: *Interpretation as Abduction.* Proceedings of ACL. 1988 http://acl.ldc.upenn.edu/P/P88/P88-1012.pdf

[Hobbs et al. 1993] Jerry R. Hobbs, Mark Stickel, Douglas Appelt and Paul Martin: *Interpretation as Abduction.* In: *Artificial Intelligence*, Vol. 63, Nos. 1-2, pp. 69-142. 1993. http://www.isi.edu/%7Ehobbs/interp-abduct-ai.pdf

[Hobbs 1995] Jerry R. Hobbs: *Deep Lexical Semantics or Commonsense Knowledge and Lexical Semantics* (Lecture). Information Sciences Institute, University of Southern California. 1995. http://www.isi.edu/~pantel/Download/Teaching/2005/CS599/cs599_Fall2005_Lecture_08.ppt

[Raina et al. 2005] Rajat Raina, Andrew Y. Ng and Christopher D. Manning: *Robust textual inference via learning and abductive reasoning.* Twentieth National Conference on Artificial Intelligence (AAAI). 2005. http://nlp.stanford.edu/~manning/papers/aaai05-learnabduction.pdf

[Wikipedia]

http://en.wikipedia.org/wiki/Abductive_reasoning

http://de.wikipedia.org/wiki/Abduktion_%28Wissenschaftstheorie%29