Computational Semantics and Recognizing Textual Entailment

Pascual Martínez-Gómez[1] Koji Mineshima[2,4]
Yusuke Miyao[1,3,4] Daisuke Bekki[1,2,3,4]

[1]AIST-AIRC, Japan
[2]Ochanomizu University, Japan
[3]National Institute of Informatics, Japan
[4]CREST, JST, Japan

August 19, 2016

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many application areas (Question Answering, Machine Translation, etc.)

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many application areas (Question Answering, Machine Translation, etc.)

- relevant factors:
  1. syntax
  2. logical words: *most, not, some, every*

Logical/ Compositional semantics

Recognizing Textual Entailment

- Does **Premise P** entail **Hypothesis H**?

P Smoking in restaurants is prohibited by law in most cities in Japan.
H Smoking in public spaces is not allowed in some cities.

Yes (Entailment)

- *The best way of testing an NLP system's semantic capacity* (Cooper et al. 1996)
- Many application areas (Question Answering, Machine Translation, etc.)

- relevant factors:

1. syntax
2. logical words: *most*, *not*, *some*, *every*

Logical/ Compositional semantics

3. content words:

restaurant $\rightarrow$ *public_space*
prohibited $\rightarrow$ $\neg$ *allowed*

Lexical Knowledge

Logic-based approaches to entailment

Natural Logic

- formalizes inferences with surface form
- ▲ only allows single premise inferences (mononicity inference)

more efficient
less expressive

MacCartney (2009)

First-order logic (FOL)

- efficient provers exist
- dominate computational linguistics
- ▲ limited expressive power

Boxer (Bos 2008)

Higher-order logic (HOL)

- high expressive power
- dominate formal semantics
- ▲ no general-purpose efficient prover exists

less efficient
more expressive

Our approach

Logic-based approaches to entailment

Natural Logic	First-order logic (FOL)	Higher-order logic (HOL)
• formalizes inferences with surface form ▲ only allows single premise inferences (mononicity inference)	• efficient provers exist • dominate computational linguistics ▲ limited expressive power	• high expressive power • dominate formal semantics ▲ no general-purpose efficient prover exists
more efficient less expressive	⟷	less efficient more expressive
MacCartney (2009)	Boxer (Bos 2008)	**Our approach**

Logic-based approaches to entailment

Natural Logic	First-order logic (FOL)	Higher-order logic (HOL)
• formalizes inferences with surface form ▲ only allows single premise inferences (mononicity inference)	• efficient provers exist • dominate computational linguistics ▲ limited expressive power	• high expressive power • dominate formal semantics ▲ no general-purpose efficient prover exists
more efficient less expressive	⟷	less efficient more expressive
MacCartney (2009)	Boxer (Bos 2008)	**Our approach**

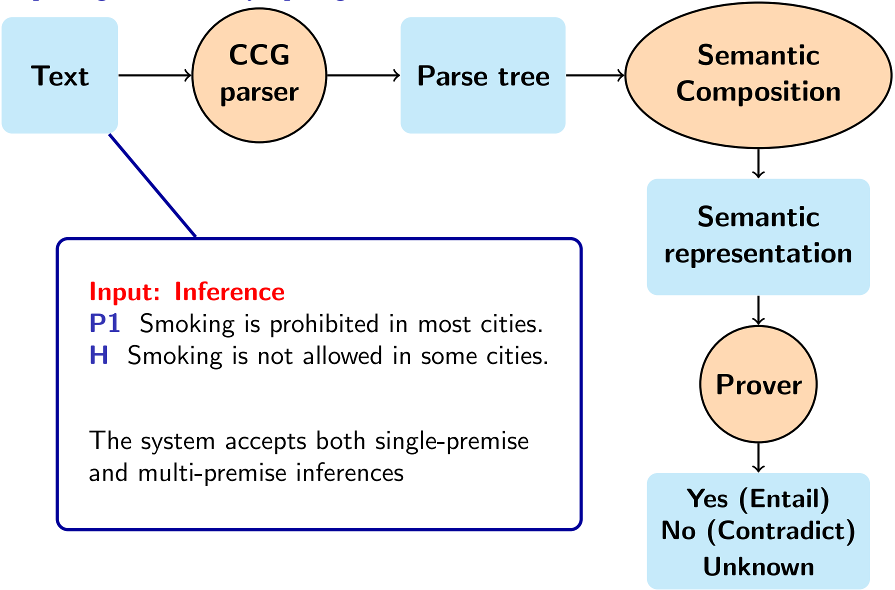
Logic-based approaches to entailment

Natural Logic	First-order logic (FOL)	Higher-order logic (HOL)
• formalizes inferences with surface form ▲ only allows single premise inferences (mononicity inference)	• efficient provers exist • dominate computational linguistics ▲ limited expressive power	• high expressive power • dominate formal semantics ▲ no general-purpose efficient prover exists
more efficient less expressive	⟷	less efficient more expressive
MacCartney (2009)	Boxer (Bos 2008)	**Our approach**

Goal: To develop a higher-order inference system specialized for natural language inferences, and combine it with a wide-coverage parser

Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda



Input: Inference

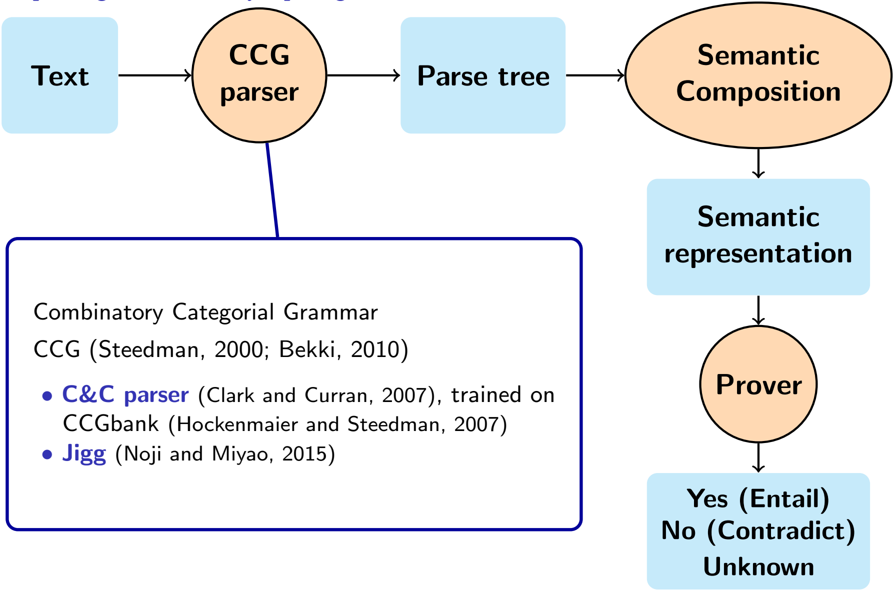
P1 Smoking is prohibited in most cities.

H Smoking is not allowed in some cities.

The system accepts both single-premise and multi-premise inferences

Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda



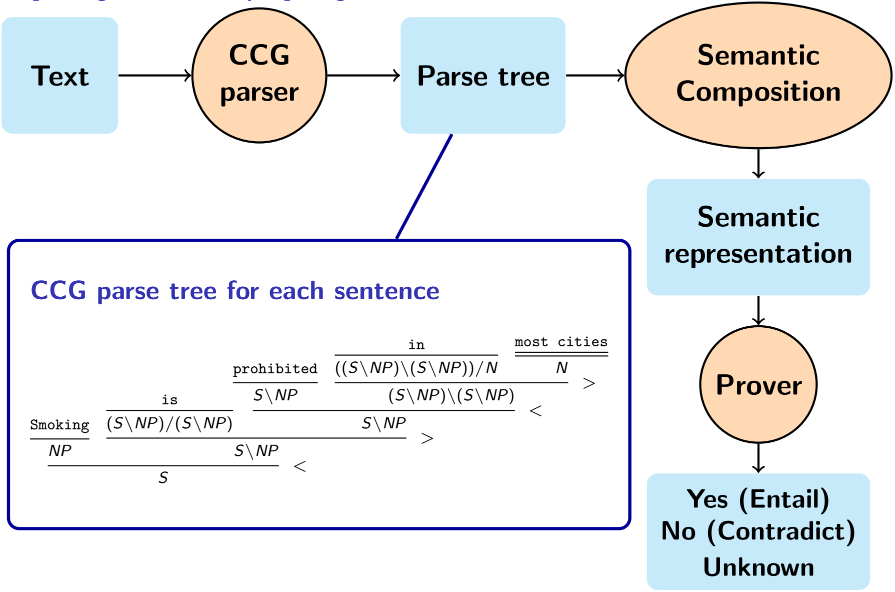
Combinatory Categorial Grammar

CCG (Steedman, 2000; Bekki, 2010)

- **C&C parser** (Clark and Curran, 2007), trained on CCGbank (Hockenmaier and Steedman, 2007)
- **Jigg** (Noji and Miyao, 2015)

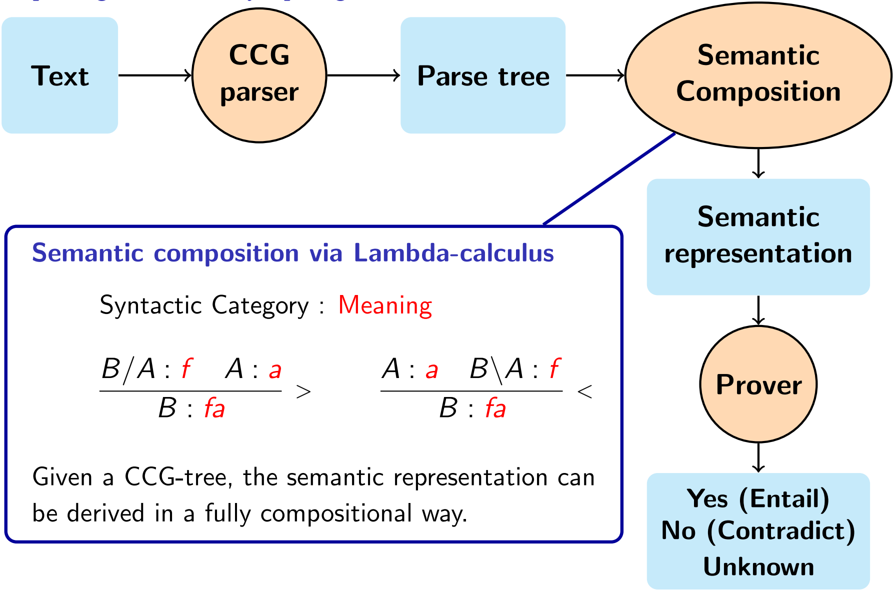
Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda



Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda



Semantic composition via Lambda-calculus

Syntactic Category : Meaning

$$\frac{B/A : f \quad A : a}{B : fa} > \qquad \frac{A : a \quad B\backslash A : f}{B : fa} <$$

Given a CCG-tree, the semantic representation can be derived in a fully compositional way.

Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda

Text → CCG parser → Parse tree → Semantic Composition → Semantic representation → Prover → Yes (Entail) / No (Contradict) / Unknown

Semantic representation in HOL

P1 Smoking is prohibited in most cities.

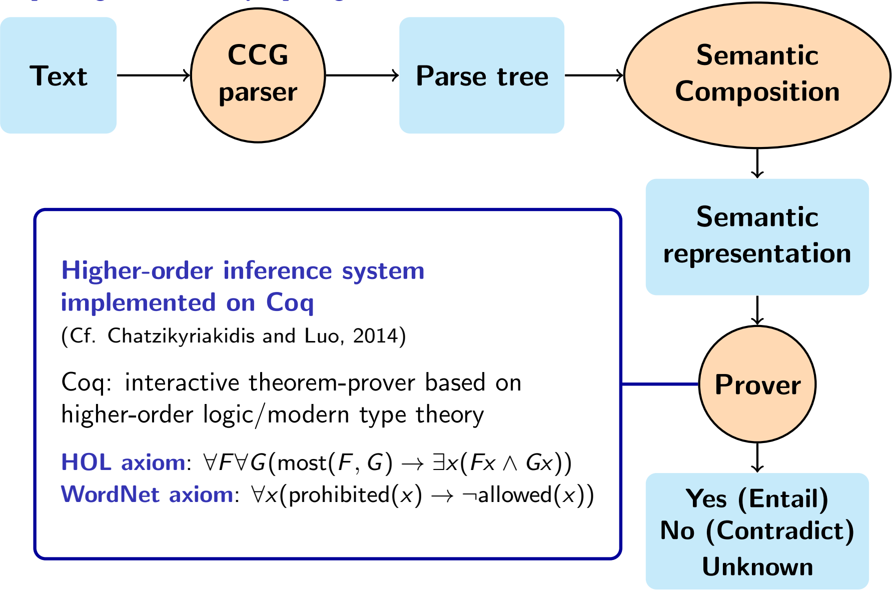
$\exists x(\mathrm{smoking}(x) \wedge$
$\quad \mathrm{most}(\lambda y.\mathrm{city}(y), \lambda y.\mathrm{prohibited}(x) \wedge \mathrm{in}(\mathrm{x}, \mathrm{y})))$

H Smoking is not allowed in some cities.

$\exists x(\mathrm{smoking}(x) \wedge$
$\quad \exists y(\mathrm{city}(y) \wedge \neg \mathrm{allowed}(x) \wedge \mathrm{in}(\mathrm{x}, \mathrm{y})))$

Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda



Higher-order inference system implemented on Coq

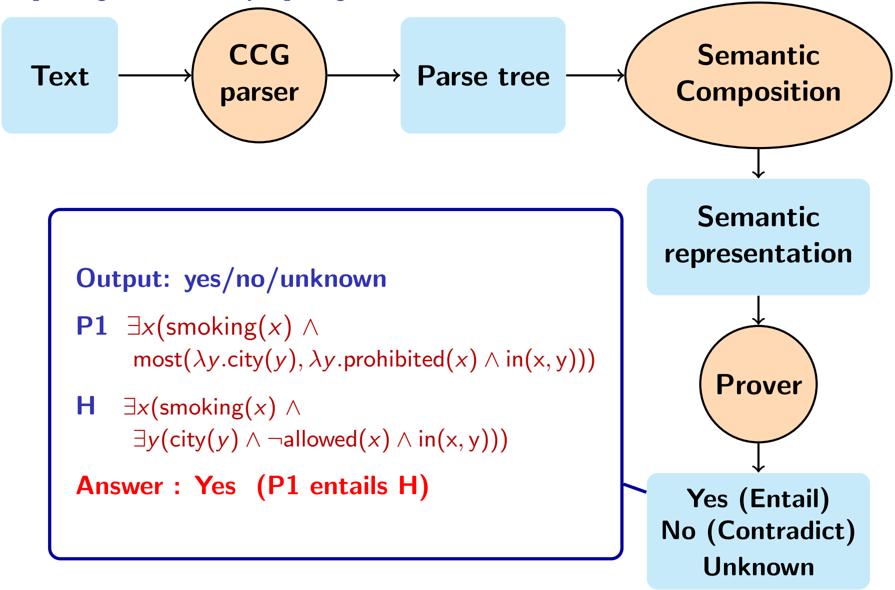
(Cf. Chatzikyriakidis and Luo, 2014)

Coq: interactive theorem-prover based on higher-order logic/modern type theory

HOL axiom: $\forall F \forall G(\text{most}(F, G) \to \exists x(Fx \wedge Gx))$
WordNet axiom: $\forall x(\text{prohibited}(x) \to \neg\text{allowed}(x))$

Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda



Output: yes/no/unknown

P1 $\exists x(\text{smoking}(x) \wedge \text{most}(\lambda y.\text{city}(y), \lambda y.\text{prohibited}(x) \wedge \text{in}(x, y)))$

H $\exists x(\text{smoking}(x) \wedge \exists y(\text{city}(y) \wedge \neg\text{allowed}(x) \wedge \text{in}(x, y)))$

Answer : Yes (P1 entails H)

Semantic composition on CCG tree

$$
\begin{array}{ccccc}
\text{Every} & \text{student} & \text{who} & \text{works} & \\
NP/N & N & (NP\backslash NP)/(S\backslash NP) & S\backslash NP & \\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{NP \quad >} & \multicolumn{2}{c}{NP\backslash NP \quad >} & \text{comes} \\
\cline{1-4}
\multicolumn{4}{c}{NP \quad <} & S\backslash NP \\
\hline
\multicolumn{5}{c}{S \quad <}
\end{array}
$$

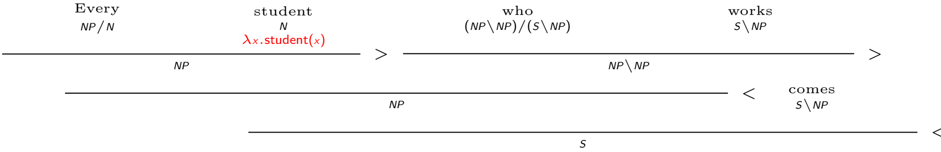
- Syntactic categories and rules indicate composition.

Semantic composition on CCG tree

$$
\begin{array}{ccccc}
\text{Every} & \text{student} & \text{who} & \text{works} & \\
NP/N & N & (NP\backslash NP)/(S\backslash NP) & S\backslash NP & \\
 & {\color{red}(\lambda Ex.E(x))(\text{student})} & & & \\
\hline
\multicolumn{2}{c}{NP \quad >} & \multicolumn{2}{c}{NP\backslash NP \quad >} & \\
\multicolumn{4}{c}{NP \quad <} & \begin{array}{c}\text{comes}\\ S\backslash NP\end{array} \\
\multicolumn{5}{c}{S \quad <}
\end{array}
$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.

Semantic composition on CCG tree



- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.

Semantic composition on CCG tree

Every NP/N ; student N $\lambda x.\mathsf{student}(x)$ $\Rightarrow_{>}$ NP

who $(NP\backslash NP)/(S\backslash NP)$; works $S\backslash NP$ $\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{work}(x))$ $\Rightarrow_{>}$ $NP\backslash NP$

NP ; $NP\backslash NP$ $\Rightarrow_{<}$ NP

NP ; comes $S\backslash NP$ $\lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{come}(x))$ $\Rightarrow_{<}$ S

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.

Semantic composition on CCG tree

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\text{Every}\\ NP/N\\ \lambda FGH.\forall x(Fx \wedge Gx \rightarrow Hx)\end{array} \quad \begin{array}{c}\text{student}\\ N\\ \lambda x.\mathsf{student}(x)\end{array}}{NP}> \quad \dfrac{\begin{array}{c}\text{who}\\ (NP\backslash NP)/(S\backslash NP)\\ \lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \wedge Fx))\end{array} \quad \begin{array}{c}\text{works}\\ S\backslash NP\\ \lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{work}(x))\end{array}}{NP\backslash NP}>}{NP}< \quad \begin{array}{c}\text{comes}\\ S\backslash NP\\ \lambda Q.Q(\lambda x.\mathsf{True})(\lambda x.\mathsf{come}(x))\end{array}}{S}<$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.

Semantic composition on CCG tree

$$\dfrac{\dfrac{\underset{\lambda FGH.\forall x(Fx \wedge Gx \rightarrow Hx)}{\underset{NP/N}{\text{Every}}} \quad \underset{\lambda x.\text{student}(x)}{\underset{N}{\text{student}}}}{\underset{\lambda GH.\forall x(\text{student}(x) \wedge Gx \rightarrow Hx)}{NP}}> \quad \dfrac{\underset{\lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \wedge Fx))}{\underset{(NP\backslash NP)/(S\backslash NP)}{\text{who}}} \quad \underset{\lambda Q.Q(\lambda x.\text{True})(\lambda x.\text{work}(x))}{\underset{S\backslash NP}{\text{works}}}}{NP\backslash NP}>}{NP}< \quad \underset{\lambda Q.Q(\lambda x.\text{True})(\lambda x.\text{come}(x))}{\underset{S\backslash NP}{\text{comes}}}$$

$$\dfrac{NP \quad S\backslash NP}{S}<$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.
- Semantic composition from leaves to root.

Semantic composition on CCG tree

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\text{Every}\\ NP/N\\ \lambda FGH.\forall x(Fx \wedge Gx \rightarrow Hx)\end{array} \quad \begin{array}{c}\text{student}\\ N\\ \lambda x.\mathrm{student}(x)\end{array}}{\begin{array}{c}NP\\ \lambda GH.\forall x(\mathrm{student}(x) \wedge Gx \rightarrow Hx)\end{array}}> \quad \dfrac{\begin{array}{c}\text{who}\\ (NP\backslash NP)/(S\backslash NP)\\ \lambda VQF.Q(\lambda x.(V(\lambda GH.Hx) \wedge Fx))\end{array} \quad \begin{array}{c}\text{works}\\ S\backslash NP\\ \lambda Q.Q(\lambda x.\mathrm{True})(\lambda x.\mathrm{work}(x))\end{array}}{\begin{array}{c}NP\backslash NP\\ \lambda QF.Q(\lambda x.(\mathrm{work}(x) \wedge Fx))\end{array}}>}{\begin{array}{c}NP\\ \lambda FH.\forall x(\mathrm{student}(x) \wedge \mathrm{work}(x) \wedge Fx \rightarrow Hx)\end{array}}< \quad \begin{array}{c}\text{comes}\\ S\backslash NP\\ \lambda Q.Q(\lambda x.\mathrm{True})(\lambda x.\mathrm{come}(x))\end{array}}{\begin{array}{c}S\\ \forall x(\mathrm{student}(x) \wedge \mathrm{work}(x) \wedge \mathrm{True} \rightarrow \mathrm{come}(x))\end{array}}<$$

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- β-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.
- Semantic composition from leaves to root.
- Logical meaning representation of the sentence at the root.

Lexical entries

❶ For **closed words**: lexical entries directly assigned to surface form (a limited number of grammatical and logical expressions): 80 entries

Example

- **category**: NP/N
- **semantics**: $\lambda F \lambda G \lambda H.\forall x(Fx \wedge Gx \rightarrow H)$
- **surf**: every

❷ For **open words**: schematic lexical entry (semantic templates) assigned to syntactic categories: 57 entries

Example

- **category**: N
- **semantics**: $\lambda E \lambda x.\mathrm{E}(x)$

"E" is a position in which a particular lexical item appears.

HOL as representation language

Higher-order constructions in natural languages

1. Generalized quantifiers
   Most students work $\rightsquigarrow$ $\mathrm{most}(\lambda.\mathrm{student}(x), \lambda x.\mathrm{work}(x))$
2. Modals
   John might come $\rightsquigarrow$ $\mathrm{might}(\mathrm{come}(j))$
3. Veridical and anti-veridical predicates
   Someone managed to come $\rightsquigarrow$ $\exists x(\mathrm{manage}(x, \mathrm{come}(x)))$
   Someone failed to come $\rightsquigarrow$ $\exists x(\mathrm{fail}(x, \mathrm{come}(x)))$
4. Attitude verbs
   John knows that some student came. $\rightsquigarrow$
   $$\mathrm{know}(\mathrm{j}, \exists x(\mathrm{student}(x) \wedge \mathrm{come}(x)))$$

- Alternative: first-order decomposition/reification (Hobbs, 1985)

Axioms for non-first-order constructions

Inference pattern	**Axiom**
Existential import	$\forall F\,\forall G\,(\mathsf{most}(F,G) \rightarrow \exists x(Fx \wedge Gx))$
Conservativity	$\forall F\,\forall G\,(\mathsf{most}(F,G) \rightarrow \mathsf{most}(F, \lambda x.(Fx \wedge Gx)))$
Monotonicity (right-upward)	$\forall F\,\forall G\,\forall H\,(\mathsf{most}(F,G) \rightarrow (\forall x(Gx \rightarrow Hx) \rightarrow \mathsf{most}(F,H)))$
Veridicality	$\forall P(\mathsf{true}(P) \rightarrow P)$ $\forall x \forall P(\mathsf{manage}(x,P) \rightarrow P)$ $\forall x \forall P(\mathsf{know}(x,P) \rightarrow P)$
Anti-veridicality	$\forall P(\mathsf{false}(P) \rightarrow \neg P)$ $\forall x \forall P(\mathsf{fail}(x,P) \rightarrow \neg P)$

Experiments on the FraCaS test suite

- The FraCaS test suite (Cooper et al., 1994): the textual inference problems to test theories of formal and computational semantics
- Most problems do not require lexical/world knowledge, but contain linguistically challenging problems.
- Three types of answer: yes, no, unknown
- Single-premise problems (55%) and multiple-premise problems (45%)

Example

fracas-026 answer: **yes** (the premises entail the hypothesis)
P1 Most Europeans are resident in Europe.
P2 All Europeans are people.
P3 All people who are resident in Europe can travel freely within Europe.
H Most Europeans can travel freely within Europe.

fracas-038 answer: **no** (the premise contradicts the hypothesis)
P1 No delegate finished the report.
H Some delegate finished the report on time.

Results: accuracy and speed

- Nutcracker = C&C parser + Boxer* + FOL prover (*bliksem*)
  + FOL model-builder (*mice*) + WordNet

  * disabled the option to use modal semantics (it didn't improve the results)

Accuracy

Section	#	Ours	Nut	L & S 13	Tian 14
Quantifiers	74	.77	.53	.62	**.80**
Plurals	33	**.67**	.52	–	–
Adjectives	22	**.68**	.32	–	–
Comparatives	31	**.48**	.45	–	–
Verbs	8	.62	.62	–	–
Attitudes	13	**.77**	.46	–	–
Total	181	**.69***	.50	–	–

* Total accuracy drops to 59% when ablating the higher-order rules

- L &S 13 = Lewis and Steedman (2013): CCG + FOL prover
- Tian 14 = Tian et al. (2014): DCS-based inference system

Results: accuracy and speed

- Nutcracker = C&C parser + Boxer* + FOL prover (*bliksem*)
  + FOL model-builder (*mice*) + WordNet
  * disabled the option to use modal semantics (it didn't improve the results)

Accuracy

Section	#	Ours	Nut	L & S 13	Tian 14
Quantifiers	74	.77	.53	.62	**.80**
Plurals	33	**.67**	.52	–	–
Adjectives	22	**.68**	.32	–	–
Comparatives	31	**.48**	.45	–	–
Verbs	8	.62	.62	–	–
Attitudes	13	**.77**	.46	–	–
Total	181	**.69***	.50	–	–

* Total accuracy drops to 59% when ablating the higher-order rules

- L &S 13 = Lewis and Steedman (2013): CCG + FOL prover
- Tian 14 = Tian et al. (2014): DCS-based inference system

Speed

Parsing and inference	sec./problem
CCG Parsing (C &C parser)	3.76
Our system with higher-order inference	**3.72**
Our system w/o higher-order inference	3.46
Nutcracker with first-order inference (first-order prover + model builder)	11.23

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

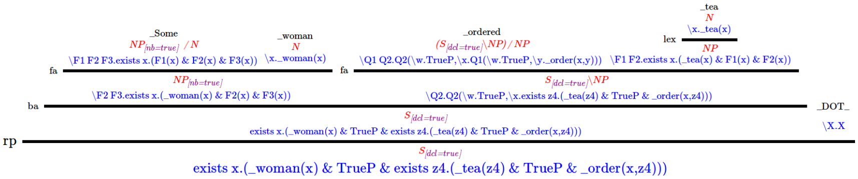
- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`

```xml
<?xml version='1.0' encoding='utf-8'?>
<root>
  <document>
    <sentences>
      <sentence>
        <tokens>
          <token id="t0_0" pos="DT" cat="NP[nb]/N"    surf="Some"   base="some" />
          <token id="t0_1" pos="NN" cat="N"           surf="woman"  base="woman" />
          <... />
        </tokens>
        <ccg root="s0_sp0" id="s0_ccg0">
          <span id="s0_sp0" child="s0_sp1 s0_sp9" category="S[dcl=true]"      rule="rp" />
          <span id="s0_sp1" child="s0_sp2 s0_sp5" category="S[dcl=true]"      rule="ba" />
          <...>
        </ccg>
        <semantics status="success" root="s0_sp0">
          <span id="s0_sp0" child="s0_sp1 s0_sp9"
                sem="exists x.(_woman(x) & TrueP & exists z1.(_tea(z1) & TrueP & _order(x,z1)))" />
          <span id="s0_sp4" type="_woman : Entity -> Prop"
                sem="\x._woman(x)" />
          <...>
        </semantics>
      </sentence>
    </sentences>
  </document>
</root>
```

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`



ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`
  - `# python prove.py semantics.xml`

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`
  - `# python prove.py semantics.xml`
- Easy to extend (declarative).

– `semantics` : λ-formula

`category` : $\text{syntactic\_category}$

$\mathtt{cond}_2$: value_2

$\mathtt{cond}_\mathrm{i}$: value_i

ccg2lambda: a few more words

https://github.com/mynlp/ccg2lambda

- Publicly available and open-sourced.
- Easy to use (simple programs):
  - `# python semparse.py ccgtrees.xml templates.yaml semantics.xml`
  - `# python visualize.py semantics.xml > semantics.html`
  - `# python prove.py semantics.xml`
- Easy to extend (declarative).
- Easy to process (XML output).

Introducing Lexical Knowledge

Introducing Lexical Knowledge

Motivation

- Does **Premise P** entail **Hypothesis H**?

P Men are sawing logs.
H Men are cutting wood.

Yes (Entailment)

Introducing Lexical Knowledge

Motivation

- Does **Premise P** entail **Hypothesis H**?

P Men are sawing logs.

H Men are cutting wood.

Yes (Entailment)

- With the knowledge:
  1. $\forall v.\mathrm{saw}(v) \rightarrow \mathrm{cut}(v)$
  2. $\forall x.\mathrm{log}(x) \rightarrow \mathrm{wood}(x)$

Introducing Lexical Knowledge

Motivation

- Does **Premise P** entail **Hypothesis H**?

P Men are sawing logs.
H Men are cutting wood.

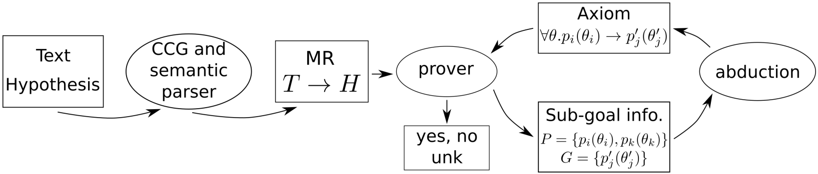
Yes (Entailment)

- With the knowledge:
  1. $\forall v.\mathrm{saw}(v) \rightarrow \mathrm{cut}(v)$
  2. $\forall x.\mathrm{log}(x) \rightarrow \mathrm{wood}(x)$

- Lack of lexical knowledge obstructs applicability of formal semantics to real tasks.

Introducing Lexical Knowledge

Approach



- Introduce knowledge at inference stage.
- Only if needed (on-demand).
- Lexical correspondences as axioms.
- Incremental.

Introducing Lexical Knowledge

Related work

- Nutcracker uses WordNet and Paraphrase Database (PPDB).
  - We contrast WordNet and a similarity on word embeddings.
- Pavlick et. al (2015) use classifier to signal entailing relations.
  - We assess entailing relations only when needed.
- Beltagy et. al (2013) use Boxer and distributional similarity scores.
  - We restrict our candidates by using logic clues.
- Abzianidze (2013) uses CCG, natural logic tableaux and WordNet.
  - We try to compensate missing knowledge using word embeddings.
- Tian et. al (2014) use their own logics, but otherwise similar to ours.

Introducing Lexical Knowledge

Abduction mechanism

- Use natural deduction as a proof calculus (prove $T \rightarrow H$).
- Decompose Text T and Hypothesis H into:
  - $P = \{p_0(\theta_0), \ldots, p_n(\theta_n)\}$, a pool of logical premises.
  - $G = \{p'_0(\theta'_0), \ldots, p'_m(\theta'_m)\}$, a pool of logical sub-goals.

Introducing Lexical Knowledge

Abduction mechanism

- Use natural deduction as a proof calculus (prove $T \rightarrow H$).
- Decompose Text T and Hypothesis H into:
  - $P = \{p_0(\theta_0), \ldots, p_n(\theta_n)\}$, a pool of logical premises.
  - $G = \{p'_0(\theta'_0), \ldots, p'_m(\theta'_m)\}$, a pool of logical sub-goals.
- Example:
  - T: "a boy eats an apple",
  - H: "a boy eats a fruit",

Introducing Lexical Knowledge

Abduction mechanism

- Use natural deduction as a proof calculus (prove $T \rightarrow H$).
- Decompose Text T and Hypothesis H into:
  - $P = \{p_0(\theta_0), \ldots, p_n(\theta_n)\}$, a pool of logical premises.
  - $G = \{p'_0(\theta'_0), \ldots, p'_m(\theta'_m)\}$, a pool of logical sub-goals.
- Example:
  - T: "a boy eats an apple", $\exists xy.\text{boy}(x) \wedge \text{apple}(y) \wedge \text{eat}(x, y)$
  - H: "a boy eats a fruit", $\exists xy.\text{boy}(x) \wedge \text{fruit}(y) \wedge \text{eat}(x, y)$

Introducing Lexical Knowledge

Abduction mechanism

- Use natural deduction as a proof calculus (prove $T \rightarrow H$).
- Decompose Text T and Hypothesis H into:
    - $P = \{p_0(\theta_0), \ldots, p_n(\theta_n)\}$, a pool of logical premises.
    - $G = \{p'_0(\theta'_0), \ldots, p'_m(\theta'_m)\}$, a pool of logical sub-goals.
- Example:
    - T: "a boy eats an apple", $\exists xy.\mathrm{boy}(x) \wedge \mathrm{apple}(y) \wedge \mathrm{eat}(x, y)$
    - H: "a boy eats a fruit", $\exists xy.\mathrm{boy}(x) \wedge \mathrm{fruit}(y) \wedge \mathrm{eat}(x, y)$
    - $P = \{\mathrm{boy}(x), \mathrm{eat}(x, y), \mathrm{apple}(y)\}$
    - $G = \{\mathrm{boy}(x'), \mathrm{eat}(x', y'), \mathrm{fruit}(y')\}$

Introducing Lexical Knowledge

Abduction mechanism

- Use natural deduction as a proof calculus (prove $T \rightarrow H$).
- Decompose Text T and Hypothesis H into:
  - $P = \{p_0(\theta_0), \ldots, p_n(\theta_n)\}$, a pool of logical premises.
  - $G = \{p'_0(\theta'_0), \ldots, p'_m(\theta'_m)\}$, a pool of logical sub-goals.
- Example:
  - T: "a boy eats an apple", $\exists xy.\mathrm{boy}(x) \wedge \mathrm{apple}(y) \wedge \mathrm{eat}(x, y)$
  - H: "a boy eats a fruit", $\exists xy.\mathrm{boy}(x) \wedge \mathrm{fruit}(y) \wedge \mathrm{eat}(x, y)$
  - $P = \{\mathrm{boy}(x), \mathrm{eat}(x, y), \mathrm{apple}(y)\}$
  - $G = \{\mathrm{boy}(x'), \mathrm{eat}(x', y'), \mathrm{fruit}(y')\}$
- Select premises and sub-goals sharing arguments: apple - fruit.
- Test linguistic relations either with WordNet or word2vec.

Introducing Lexical Knowledge
Linguistic Relations

Using WordNet

- Relations and corresponding axioms:
  - synonymy (e.g. house $\rightarrow$ home),
  - hypernymy (e.g. sea $\rightarrow$ water),
  - adjectival similarity (e.g. huge $\rightarrow$ big),
  - derivationally related forms (accommodating $\rightarrow$ accommodation),
  - inflection relations (e.g. wooded $\rightarrow$ wood),
  - antonymy relations (e.g. big $\rightarrow \neg$small).

Introducing Lexical Knowledge

Linguistic Relations

Using WordNet

- Relations and corresponding axioms:
  - synonymy (e.g. house $\rightarrow$ home),
  - hypernymy (e.g. sea $\rightarrow$ water),
  - adjectival similarity (e.g. huge $\rightarrow$ big),
  - derivationally related forms (accommodating $\rightarrow$ accommodation),
  - inflection relations (e.g. wooded $\rightarrow$ wood),
  - antonymy relations (e.g. big $\rightarrow \neg$small).

Using word2vec

- Compute cosine similarity between source and target words.
- If similarity is beyond a threshold, then insert entailing axiom.

Introducing Lexical Knowledge

Experiments: Setup

Dataset: SICK, test split

Basic statistics	Test
Number of problems	4,927
Percentage yes/no/unk	.29/.15/.57
Running words	105,040
Vocabulary	2,269
Avg. premise length	10.7
Avg. conclusion length	10.5
Avg. OoV w.r.t. T	3.62
Avg. OoV w.r.t. H	3.76

Other conditions:

- C&C and EasyCCG derivations.
- ccg2lambda for composition: `github.com/mynlp/ccg2lambda`
- Event semantics, Coq prover.

Introducing Lexical Knowledge

Experiments: Setup

Some examples of problems

Problem ID	T-H pairs	Entailment
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes
4114	T: *There is no man eating food .* H: *A man is eating a pizza .*	No
718	T: *A few men in a competition are running outside .* H: *A few men are running competitions outside .*	Unknown

Introducing Lexical Knowledge

Experiments: Results

System	Precision	Recall	Accuracy
Illinois-LH	81.56	81.87	84.57
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32

Introducing Lexical Knowledge

Experiments: Results

System	Precision	Recall	Accuracy
Illinois-LH	81.56	81.87	84.57
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32
The Meaning Factory	93.63	60.64	81.60
LangPro Hybrid-800	97.95	58.11	81.35
Nutcracker + WN, PPDB	–	–	78.60
Nutcracker + WN	–	–	77.50
Nutcracker	–	–	74.30

Introducing Lexical Knowledge

Experiments: Results

System	Precision	Recall	Accuracy
Illinois-LH	81.56	81.87	84.57
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32
The Meaning Factory	93.63	60.64	81.60
LangPro Hybrid-800	97.95	58.11	81.35
Nutcracker + WN, PPDB	–	–	78.60
Nutcracker + WN	–	–	77.50
Nutcracker	–	–	74.30
Baseline (majority)	–	–	56.69

Introducing Lexical Knowledge

Experiments: Results

System	Precision	Recall	Accuracy
Illinois-LH	81.56	81.87	84.57
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32
The Meaning Factory	93.63	60.64	81.60
LangPro Hybrid-800	97.95	58.11	81.35
Nutcracker + WN, PPDB	–	–	78.60
Nutcracker + WN	–	–	77.50
Nutcracker	–	–	74.30
Baseline (majority)	–	–	56.69
No abduction	99.09	46.34	76.63

Introducing Lexical Knowledge

Experiments: Results

System	Precision	Recall	Accuracy
Illinois-LH	81.56	81.87	84.57
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32
The Meaning Factory	93.63	60.64	81.60
LangPro Hybrid-800	97.95	58.11	81.35
Nutcracker + WN, PPDB	–	–	78.60
Nutcracker + WN	–	–	77.50
Nutcracker	–	–	74.30
Baseline (majority)	–	–	56.69
No abduction	99.09	46.34	76.63
SPSA WordNet	97.50	62.32	83.05
SPSA word2vec@0.4	83.54	60.55	77.80

- Effective use of WordNet with SPSA, increasing accuracy by 6.5%.
    - Use of WordNet is more effective than other systems (e.g. Nutcracker).
- SPSA method minimizes false positives, by restricting candidates.

Introducing Lexical Knowledge

Error analysis: WordNet

Prob. ID	T-H pairs	Gold	System	Axioms
True positives $\times 32$ (w.r.t to no-abduction system)				
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathrm{saw}(v) \rightarrow \mathrm{cut}(v)$ $\forall x.\mathrm{log}(x) \rightarrow \mathrm{wood}(x)$
211	T: *Two dogs are playing by a tree .* H: *Two dogs are playing by a plant .*	Yes	Yes	$\forall x.\mathrm{tree}(x) \rightarrow \mathrm{plant}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathrm{slice}(v) \rightarrow \mathrm{cut}(v)$
Others: tree/plant, eat/devour, lady/woman, barbell/weight, pencil/cosmetic, place/put, talk/speak, lemon/fruit, canoe/boat, large/big, woman/people, dark/dim, young/little, grass/grassy, field/area, hurl/throw, pizza/food, bounce/jump, draw/drawing, banana/fruit, man/person, pause/stop, street/road, dog/animal, ocean/water, big/huge,				

Introducing Lexical Knowledge

Error analysis: WordNet

Prob. ID	T-H pairs	Gold	System	Axioms
True positives ×32 (w.r.t to no-abduction system)				
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathrm{saw}(v) \rightarrow \mathrm{cut}(v)$ $\forall x.\mathrm{log}(x) \rightarrow \mathrm{wood}(x)$
211	T: *Two dogs are playing by a tree .* H: *Two dogs are playing by a plant .*	Yes	Yes	$\forall x.\mathrm{tree}(x) \rightarrow \mathrm{plant}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathrm{slice}(v) \rightarrow \mathrm{cut}(v)$
Others: tree/plant, eat/devour, lady/woman, barbell/weight, pencil/cosmetic, place/put, talk/speak, lemon/fruit, canoe/boat, large/big, woman/people, dark/dim, young/little, grass/grassy, field/area, hurl/throw, pizza/food, bounce/jump, draw/drawing, banana/fruit, man/person, pause/stop, street/road, dog/animal, ocean/water, big/huge,				
False positives ×2 (w.r.t to no-abduction system)				
530	T: *A biker is wearing gear which is black .* H: *A biker wearing black is breaking the gears .*	Unk	Yes	$\forall v.\mathrm{wear}(v) \rightarrow \mathrm{break}(v)$

Introducing Lexical Knowledge

Error analysis: WordNet

Prob. ID	T-H pairs	Gold	System	Axioms
True positives $\times 32$ (w.r.t to no-abduction system)				
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathsf{saw}(v) \rightarrow \mathsf{cut}(v)$ $\forall x.\mathsf{log}(x) \rightarrow \mathsf{wood}(x)$
211	T: *Two dogs are playing by a tree .* H: *Two dogs are playing by a plant .*	Yes	Yes	$\forall x.\mathsf{tree}(x) \rightarrow \mathsf{plant}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathsf{slice}(v) \rightarrow \mathsf{cut}(v)$
Others: tree/plant, eat/devour, lady/woman, barbell/weight, pencil/cosmetic, place/put, talk/speak, lemon/fruit, canoe/boat, large/big, woman/people, dark/dim, young/little, grass/grassy, field/area, hurl/throw, pizza/food, bounce/jump, draw/drawing, banana/fruit, man/person, pause/stop, street/road, dog/animal, ocean/water, big/huge,				
False positives $\times 2$ (w.r.t to no-abduction system)				
530	T: *A biker is wearing gear which is black .* H: *A biker wearing black is breaking the gears .*	Unk	Yes	$\forall v.\mathsf{wear}(v) \rightarrow \mathsf{break}(v)$
False negatives: missing word-to-word axioms $\times 12$				
1495	T: *A man is playing a guitar .* H: *A man is strumming a guitar .*	Yes	Unk	$\forall v.\mathsf{play}(v) \rightarrow \mathsf{strum}(v)$
4360	T: *A man is cutting a note .* H: *A man is cutting a paper .*	Yes	Unk	$\forall x.\mathsf{note}(x) \rightarrow \mathsf{paper}(x)$
Others: healing/helping, fitting/applying, ringers/wrestlers, lunging/jumping, drawing/tatoo, lady/girl, elder/elderly, woman/lady				

Introducing Lexical Knowledge

Error analysis: WordNet

Prob. ID	T-H pairs	Gold	System	Axioms
True positives ×32 (w.r.t to no-abduction system)				
1412	T: *Men are sawing logs .* H: *Men are cutting wood .*	Yes	Yes	$\forall v.\mathrm{saw}(v) \rightarrow \mathrm{cut}(v)$ $\forall x.\mathrm{log}(x) \rightarrow \mathrm{wood}(x)$
211	T: *Two dogs are playing by a tree .* H: *Two dogs are playing by a plant .*	Yes	Yes	$\forall x.\mathrm{tree}(x) \rightarrow \mathrm{plant}(x)$
2404	T: *The lady is slicing a tomato .* H: *There is no one cutting a tomato .*	No	No	$\forall v.\mathrm{slice}(v) \rightarrow \mathrm{cut}(v)$
Others: tree/plant, eat/devour, lady/woman, barbell/weight, pencil/cosmetic, place/put, talk/speak, lemon/fruit, canoe/boat, large/big, woman/people, dark/dim, young/little, grass/grassy, field/area, hurl/throw, pizza/food, bounce/jump, draw/drawing, banana/fruit, man/person, pause/stop, street/road, dog/animal, ocean/water, big/huge,				
False positives ×2 (w.r.t to no-abduction system)				
530	T: *A biker is wearing gear which is black .* H: *A biker wearing black is breaking the gears .*	Unk	Yes	$\forall v.\mathrm{wear}(v) \rightarrow \mathrm{break}(v)$
False negatives: missing word-to-word axioms ×12				
1495	T: *A man is playing a guitar .* H: *A man is strumming a guitar .*	Yes	Unk	$\forall v.\mathrm{play}(v) \rightarrow \mathrm{strum}(v)$
4360	T: *A man is cutting a note .* H: *A man is cutting a paper .*	Yes	Unk	$\forall x.\mathrm{note}(x) \rightarrow \mathrm{paper}(x)$
Others: healing/helping, fitting/applying, ringers/wrestlers, lunging/jumping, drawing/tatoo, lady/girl, elder/elderly, woman/lady				
False negatives: missing phrase axioms ×38				
1266	T: *A band is playing on a stage .* H: *A band is playing onstage .*	Yes	Unk	"on a stage" → "onstage"
1584	T: *A hole is being burrowed by the badger .* H: *A badger is shrewdly digging the earth .*	Yes	Unk	"burrow a hole" → "dig the earth"
Others: wooded area/woods, cut up/cut into pieces, sew with a machine/use machine made for sewing, in white/wearing white clothes, burn with a blow torch/set, fire, doing a dance/dancing, oil for cooking/cooking oil, green ball/gren colored ball, water scooter/scooter for water, tall and green grass/field, climb/climb up, ...				

Introducing Lexical Knowledge

Conclusions

Future work

- Introduce progressively features of DTS.
- Solve phrase-to-phrase axioms.

Future work

- Introduce progressively features of DTS.
- Solve phrase-to-phrase axioms.

Conclusions

- Competitive logics system is possible.
- There is a lot of potential to produce even better results.

Introducing Lexical Knowledge

Conclusions

Future work

- Introduce progressively features of DTS.
- Solve phrase-to-phrase axioms.

Conclusions

- Competitive logics system is possible.
- There is a lot of potential to produce even better results.

Thanks! `pascual.mg@aist.go.jp`