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* → *public space*
    *prohibited* → ¬*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

First-order logic (FOL)

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

Higher-order logic (HOL)

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

MacCartney (2009)  Boxer (Bos 2008)  Our approach
Logic-based approaches to entailment

**Natural Logic**
- formalizes inferences with surface form
- only allows single premise inferences (mononicity inference)
- more efficient
- less expressive

**First-order logic (FOL)**
- efficient provers exist
- dominate computational linguistics
- limited expressive power

**Higher-order logic (HOL)**
- high expressive power
- dominate formal semantics
- no general-purpose efficient prover exists
- less efficient
- more expressive

MacCartney (2009)  
Boxer (Bos 2008)  
Our approach
Logic-based approaches to entailment

**Natural Logic**
- formalizes inferences with surface form
- only allows single premise inferences (mononicity inference)

**First-order logic (FOL)**
- efficient provers exist
- dominate computational linguistics
- limited expressive power

**Higher-order logic (HOL)**
- high expressive power
- dominate formal semantics
- no general-purpose efficient prover exists

MacCartney (2009)  Boxer (Bos 2008)

**Our approach**
Logic-based approaches to entailment

**Natural Logic**
- formalizes inferences with surface form
  - only allows single premise inferences (mononicity inference)
- more efficient
- less expressive

**First-order logic (FOL)**
- efficient provers exist
- dominate computational linguistics
  - limited expressive power

**Higher-order logic (HOL)**
- high expressive power
- dominate formal semantics
  - no general-purpose efficient prover exists
- less efficient
- more expressive

---

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

MacCartney (2009)  Boxer (Bos 2008)

Our approach
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

- **Text** → **CCG parser** → **Parse tree** → **Semantic Composition** → **Semantic representation** → **Prover**

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)

Yes (Entail)
No (Contradict)
Unknown
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

CCG parse tree for each sentence

Smoking is prohibited in most cities

(S\NP)/\NP \(\frac{\((S\NP)/(S\NP)\)/N}{(S\NP)/(S\NP)}\) \(\frac{\text{in}}{N}\) \(\frac{\text{most cities}}{N}\) >

\(\frac{\text{is}}{S\NP}\)

\(\frac{\text{prohibited}}{S\NP}\)

\(\frac{\text{in}}{N}\)

\(\frac{\text{most cities}}{N}\)

\(>\)

\(\frac{\text{NP}}{S\NP}\)

\(\frac{\text{in}}{N}\)

\(>\)

\(\frac{\text{most cities}}{N}\)

\(>\)

\(\frac{\text{in}}{N}\)

\(>\)

\(\frac{\text{most cities}}{N}\)

\(<\)

\(\frac{\text{NP}}{S\NP}\)

\(\frac{\text{in}}{N}\)

\(<\)

\(\frac{\text{most cities}}{N}\)

\(<\)
Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda

Semantic composition via Lambda-calculus

Syntactic Category: Meaning

\[
\frac{B/A : f \quad A : a}{B : fa} \quad > \quad \frac{A : a \quad B \setminus A : f}{B : fa} <
\]

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

Semantic representation in HOL

P1  Smoking is prohibited in most cities.
    \exists x (\text{smoking}(x) \land 
        \text{most}(\lambda y. \text{city}(y), \lambda y. \text{prohibited}(x) \land \text{in}(x, y)))

H  Smoking is not allowed in some cities.
    \exists x (\text{smoking}(x) \land 
        \exists y (\text{city}(y) \land \neg \text{allowed}(x) \land \text{in}(x, y)))
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

**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:** \( \forall F \forall G (\text{most}(F, G) \rightarrow \exists x (Fx \land Gx)) \)

**WordNet axiom:** \( \forall x (\text{prohibited}(x) \rightarrow \neg \text{allowed}(x)) \)
Higher-order inference system: ccg2lambda

https://github.com/mynlp/ccg2lambda

Text → CCG parser → Parse tree → Semantic Composition

Output: yes/no/unknown

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

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

Answer: Yes (P1 entails H)
• Syntactic categories and rules indicate composition.
Semantic composition on CCG tree

- 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.
- $\beta$-reduction with lemmas as arguments.
Semantic composition on CCG tree

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- $\beta$-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
Semantic composition on CCG tree

- Syntactic categories and rules indicate composition.
- Open words: schematic lexical entries match syntactic categories.
- $\beta$-reduction with lemmas as arguments.
- Semantics more interesting for verbs.
- Closed words: direct assignment.
Semantic composition on CCG tree

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

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

1. 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**: $\lambda F \lambda G \lambda H. \forall x (F x \land G x \rightarrow H)$
- **surf**: every

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

Example

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

“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 \( \leadsto \) most(\( \lambda \).student(x), \( \lambda \).work(x))

2. Modals
   John might come \( \leadsto \) might(come(j))

3. Veridical and anti-veridical predicates
   Someone managed to come \( \leadsto \) \( \exists \).manage(x, come(x)))
   Someone failed to come \( \leadsto \) \( \exists \).fail(x, come(x)))

4. Attitude verbs
   John knows that some student came. \( \leadsto \)
   know(j, \( \exists \).student(x) \& come(x)))

- Alternative: first-order decomposition/reification (Hobbs, 1985)
## Axioms for non-first-order constructions

<table>
<thead>
<tr>
<th>Inference pattern</th>
<th>Axiom</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Existential import</strong></td>
<td>$\forall F \forall G (\text{most}(F, G) \rightarrow \exists x (Fx \land Gx))$</td>
</tr>
<tr>
<td><strong>Conservativity</strong></td>
<td>$\forall F \forall G (\text{most}(F, G) \rightarrow \text{most}(F, \lambda x. (Fx \land Gx)))$</td>
</tr>
<tr>
<td><strong>Monotonicity</strong> (right-upward)</td>
<td>$\forall F \forall G \forall H (\text{most}(F, G) \rightarrow (\forall x (Gx \rightarrow Hx) \rightarrow \text{most}(F, H)))$</td>
</tr>
<tr>
<td><strong>Veridicality</strong></td>
<td>$\forall P (\text{true}(P) \rightarrow P)$</td>
</tr>
<tr>
<td></td>
<td>$\forall x\forall P (\text{manage}(x, P) \rightarrow P)$</td>
</tr>
<tr>
<td></td>
<td>$\forall x\forall P (\text{know}(x, P) \rightarrow P)$</td>
</tr>
<tr>
<td><strong>Anti-veridicality</strong></td>
<td>$\forall P (\text{false}(P) \rightarrow \neg P)$</td>
</tr>
<tr>
<td></td>
<td>$\forall x\forall P (\text{fail}(x, P) \rightarrow \neg P)$</td>
</tr>
</tbody>
</table>
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**

<table>
<thead>
<tr>
<th>fraction-026</th>
<th>answer: yes (the premises entail the hypothesis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Most Europeans are resident in Europe.</td>
</tr>
<tr>
<td>P2</td>
<td>All Europeans are people.</td>
</tr>
<tr>
<td>P3</td>
<td>All people who are resident in Europe can travel freely within Europe.</td>
</tr>
<tr>
<td>H</td>
<td>Most Europeans can travel freely within Europe.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fraction-038</th>
<th>answer: no (the premise contradicts the hypothesis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>No delegate finished the report.</td>
</tr>
<tr>
<td>H</td>
<td>Some delegate finished the report on time.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Section</th>
<th>#</th>
<th>Ours</th>
<th>Nut</th>
<th>L &amp; S 13</th>
<th>Tian 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantifiers</td>
<td>74</td>
<td>.77</td>
<td>.53</td>
<td>.62</td>
<td>.80</td>
</tr>
<tr>
<td>Plurals</td>
<td>33</td>
<td>.67</td>
<td>.52</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Adjectives</td>
<td>22</td>
<td>.68</td>
<td>.32</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Comparatives</td>
<td>31</td>
<td>.48</td>
<td>.45</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Verbs</td>
<td>8</td>
<td>.62</td>
<td>.62</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Attitudes</td>
<td>13</td>
<td>.77</td>
<td>.46</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>181</td>
<td><strong>.69</strong></td>
<td>.50</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

* 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

<table>
<thead>
<tr>
<th>Section</th>
<th>#</th>
<th>Ours</th>
<th>Nut</th>
<th>L &amp; S 13</th>
<th>Tian 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantifiers</td>
<td>74</td>
<td>.77</td>
<td>.53</td>
<td>.62</td>
<td>.80</td>
</tr>
<tr>
<td>Plurals</td>
<td>33</td>
<td>.67</td>
<td>.52</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Adjectives</td>
<td>22</td>
<td>.68</td>
<td>.32</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Comparatives</td>
<td>31</td>
<td>.48</td>
<td>.45</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Verbs</td>
<td>8</td>
<td>.62</td>
<td>.62</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Attitudes</td>
<td>13</td>
<td>.77</td>
<td>.46</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Total</td>
<td>181</td>
<td>.69*</td>
<td>.50</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

* 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

<table>
<thead>
<tr>
<th>Parsing and inference</th>
<th>sec./problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCG Parsing (C &amp; C parser)</td>
<td>3.76</td>
</tr>
<tr>
<td>Our system with higher-order inference</td>
<td>3.72</td>
</tr>
<tr>
<td>Our system w/o higher-order inference</td>
<td>3.46</td>
</tr>
<tr>
<td>Nutcracker with first-order inference (first-order prover + model builder)</td>
<td>11.23</td>
</tr>
</tbody>
</table>
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`
  - `# 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
**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`
c cg2lambda: 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 : \lambda\text{-formula}
    - category : syntactic\_category
    - cond_2 : value_2
    - cond_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*.

• With the knowledge:

  1. $\forall v. \text{saw}(v) \rightarrow \text{cut}(v)$
  2. $\forall x. \text{log}(x) \rightarrow \text{wood}(x)$

**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. $\forall v. \text{saw}(v) \rightarrow \text{cut}(v)$
  2. $\forall x. \text{log}(x) \rightarrow \text{wood}(x)$

- 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.
  - 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”,

Select premises and sub-goals sharing arguments: apple - fruit. Test linguistic relations either with WordNet or word2vec.
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”, \( \exists xy. \text{boy}(x) \land \text{apple}(y) \land \text{eat}(x, y) \)
  - \( H \): “a boy eats a fruit”, \( \exists xy. \text{boy}(x) \land \text{fruit}(y) \land \text{eat}(x, y) \)

Select premises and sub-goals sharing arguments: apple - fruit. Test linguistic relations either with WordNet or word2vec.
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”, $\exists xy. \text{boy}(x) \land \text{apple}(y) \land \text{eat}(x, y)$
  - $H$: “a boy eats a fruit”, $\exists xy. \text{boy}(x) \land \text{fruit}(y) \land \text{eat}(x, y)$
  - $P = \{\text{boy}(x), \text{eat}(x, y), \text{apple}(y)\}$
  - $G = \{\text{boy}(x'), \text{eat}(x', y'), \text{fruit}(y')\}$

Select premises and sub-goals sharing arguments: apple - fruit.
Test linguistic relations either with WordNet or word2vec.
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”, $\exists xy.\text{boy}(x) \land \text{apple}(y) \land \text{eat}(x, y)$
  - $H$: “a boy eats a fruit”, $\exists xy.\text{boy}(x) \land \text{fruit}(y) \land \text{eat}(x, y)$
  - $P = \{\text{boy}(x), \text{eat}(x, y), \text{apple}(y)\}$
  - $G = \{\text{boy}(x'), \text{eat}(x', y'), \text{fruit}(y')\}$
- 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 → home),
  - hypernymy (e.g. sea → water),
  - adjectival similarity (e.g. huge → big),
  - derivationally related forms (accommodating → accommodation),
  - inflection relations (e.g. wooded → wood),
  - antonymy relations (e.g. big → ¬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 $\rightarrow$ ¬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

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

**Other conditions:**

- C&C and EasyCCG derivations.
- `ccg2lambda` for composition: [github.com/mynlp/ccg2lambda](https://github.com/mynlp/ccg2lambda)
- Event semantics, Coq prover.
Introducing Lexical Knowledge

Experiments: Setup

Some examples of problems

<table>
<thead>
<tr>
<th>Problem ID</th>
<th>T-H pairs</th>
<th>Entailment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1412</td>
<td>T: <em>Men are sawing logs</em>. H: <em>Men are cutting wood</em>.</td>
<td>Yes</td>
</tr>
<tr>
<td>4114</td>
<td>T: <em>There is no man eating food</em>. H: <em>A man is eating a pizza</em>.</td>
<td>No</td>
</tr>
<tr>
<td>718</td>
<td>T: <em>A few men in a competition are running outside</em>. H: <em>A few men are running competitions outside</em>.</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
## Introducing Lexical Knowledge

### Experiments: Results

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illinois-LH</td>
<td>81.56</td>
<td>81.87</td>
<td>84.57</td>
</tr>
<tr>
<td>ECNU</td>
<td>84.37</td>
<td>74.37</td>
<td>83.64</td>
</tr>
<tr>
<td>UNAL-NLP</td>
<td>81.99</td>
<td>76.80</td>
<td>83.05</td>
</tr>
<tr>
<td>SemantiKLUE</td>
<td>85.40</td>
<td>69.63</td>
<td>82.32</td>
</tr>
</tbody>
</table>
### Introducing Lexical Knowledge

#### Experiments: Results

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illinois-LH</td>
<td>81.56</td>
<td>81.87</td>
<td>84.57</td>
</tr>
<tr>
<td>ECNU</td>
<td>84.37</td>
<td>74.37</td>
<td>83.64</td>
</tr>
<tr>
<td>UNAL-NLP</td>
<td>81.99</td>
<td>76.80</td>
<td>83.05</td>
</tr>
<tr>
<td>SemantiKLUE</td>
<td>85.40</td>
<td>69.63</td>
<td>82.32</td>
</tr>
<tr>
<td>The Meaning Factory</td>
<td>93.63</td>
<td>60.64</td>
<td>81.60</td>
</tr>
<tr>
<td>LangPro Hybrid-800</td>
<td>97.95</td>
<td>58.11</td>
<td>81.35</td>
</tr>
<tr>
<td>Nutcracker + WN, PPDB</td>
<td>—</td>
<td>—</td>
<td>78.60</td>
</tr>
<tr>
<td>Nutcracker + WN</td>
<td>—</td>
<td>—</td>
<td>77.50</td>
</tr>
<tr>
<td>Nutcracker</td>
<td>—</td>
<td>—</td>
<td>74.30</td>
</tr>
</tbody>
</table>
Introducing Lexical Knowledge

Experiments: Results

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illinois-LH</td>
<td>81.56</td>
<td>81.87</td>
<td>84.57</td>
</tr>
<tr>
<td>ECNU</td>
<td>84.37</td>
<td>74.37</td>
<td>83.64</td>
</tr>
<tr>
<td>UNAL-NLP</td>
<td>81.99</td>
<td>76.80</td>
<td>83.05</td>
</tr>
<tr>
<td>SemantiKLUE</td>
<td>85.40</td>
<td>69.63</td>
<td>82.32</td>
</tr>
<tr>
<td>The Meaning Factory</td>
<td>93.63</td>
<td>60.64</td>
<td>81.60</td>
</tr>
<tr>
<td>LangPro Hybrid-800</td>
<td>97.95</td>
<td>58.11</td>
<td>81.35</td>
</tr>
<tr>
<td>Nutcracker + WN, PPDB</td>
<td>—</td>
<td>—</td>
<td>78.60</td>
</tr>
<tr>
<td>Nutcracker + WN</td>
<td>—</td>
<td>—</td>
<td>77.50</td>
</tr>
<tr>
<td>Nutcracker</td>
<td>—</td>
<td>—</td>
<td>74.30</td>
</tr>
<tr>
<td>Baseline (majority)</td>
<td>—</td>
<td>—</td>
<td>56.69</td>
</tr>
</tbody>
</table>
## Introducing Lexical Knowledge

### Experiments: Results

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illinois-LH</td>
<td>81.56</td>
<td>81.87</td>
<td>84.57</td>
</tr>
<tr>
<td>ECNU</td>
<td>84.37</td>
<td>74.37</td>
<td>83.64</td>
</tr>
<tr>
<td>UNAL-NLP</td>
<td>81.99</td>
<td>76.80</td>
<td>83.05</td>
</tr>
<tr>
<td>SemantiKLUE</td>
<td>85.40</td>
<td>69.63</td>
<td>82.32</td>
</tr>
<tr>
<td>The Meaning Factory</td>
<td>93.63</td>
<td>60.64</td>
<td>81.60</td>
</tr>
<tr>
<td>LangPro Hybrid-800</td>
<td>97.95</td>
<td>58.11</td>
<td>81.35</td>
</tr>
<tr>
<td>Nutcracker + WN, PPDB</td>
<td>—</td>
<td>—</td>
<td>78.60</td>
</tr>
<tr>
<td>Nutcracker + WN</td>
<td>—</td>
<td>—</td>
<td>77.50</td>
</tr>
<tr>
<td>Nutcracker</td>
<td>—</td>
<td>—</td>
<td>74.30</td>
</tr>
<tr>
<td>Baseline (majority)</td>
<td>—</td>
<td>—</td>
<td>56.69</td>
</tr>
<tr>
<td>No abduction</td>
<td>99.09</td>
<td>46.34</td>
<td>76.63</td>
</tr>
</tbody>
</table>
## Introducing Lexical Knowledge

### Experiments: Results

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

- 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

<table>
<thead>
<tr>
<th>Prob. ID</th>
<th>T-H pairs</th>
<th>Gold</th>
<th>System</th>
<th>Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1412</td>
<td>T: <em>Men are sawing logs</em>. H: <em>Men are cutting wood</em>.</td>
<td>Yes</td>
<td>Yes</td>
<td>$\forall v. \text{saw}(v) \rightarrow \text{cut}(v)$ $\forall x. \text{log}(x) \rightarrow \text{wood}(x)$</td>
</tr>
<tr>
<td>211</td>
<td>T: <em>Two dogs are playing by a tree</em>. H: <em>Two dogs are playing by a plant</em>.</td>
<td>Yes</td>
<td>Yes</td>
<td>$\forall x. \text{tree}(x) \rightarrow \text{plant}(x)$</td>
</tr>
<tr>
<td>2404</td>
<td>T: <em>The lady is slicing a tomato</em>. H: <em>There is no one cutting a tomato</em>.</td>
<td>No</td>
<td>No</td>
<td>$\forall v. \text{slice}(v) \rightarrow \text{cut}(v)$</td>
</tr>
</tbody>
</table>

True positives $\times 32$ (w.r.t to no-abduction system)

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

<table>
<thead>
<tr>
<th>Prob. ID</th>
<th>T-H pairs</th>
<th>Gold</th>
<th>System</th>
<th>Axioms</th>
</tr>
</thead>
</table>
| 1412    | **T:** Men are sawing logs .<br>**H:** Men are cutting wood .             | Yes  | Yes    | $\forall \mathbf{v}. \text{saw}(\mathbf{v}) \rightarrow \text{cut}(\mathbf{v})$
|         |                                                                          |      |        | $\forall \mathbf{x}. \text{log}(\mathbf{x}) \rightarrow \text{wood}(\mathbf{x})$         |
| 211     | **T:** Two dogs are playing by a tree .<br>**H:** Two dogs are playing by a plant . | Yes  | Yes    | $\forall \mathbf{x}. \text{tree}(\mathbf{x}) \rightarrow \text{plant}(\mathbf{x})$         |
| 2404    | **T:** The lady is slicing a tomato .<br>**H:** There is no one cutting a tomato . | No   | No     | $\forall \mathbf{v}. \text{slice}(\mathbf{v}) \rightarrow \text{cut}(\mathbf{v})$           |

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,

<table>
<thead>
<tr>
<th>False positives $\times 2$ (w.r.t to no-abduction system)</th>
</tr>
</thead>
<tbody>
<tr>
<td>530</td>
</tr>
</tbody>
</table>
## Introducing Lexical Knowledge

### Error analysis: WordNet

<table>
<thead>
<tr>
<th>Prob. ID</th>
<th>T-H pairs</th>
<th>Gold</th>
<th>System</th>
<th>Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True positives $\times 32$ (w.r.t to no-abduction system)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1412</td>
<td>T: <em>Men are sawing logs.</em></td>
<td>Yes</td>
<td>Yes</td>
<td>$\forall v.\text{saw}(v) \rightarrow \text{cut}(v)$</td>
</tr>
<tr>
<td></td>
<td>H: <em>Men are cutting wood.</em></td>
<td></td>
<td></td>
<td>$\forall x.\text{log}(x) \rightarrow \text{wood}(x)$</td>
</tr>
<tr>
<td>211</td>
<td>T: <em>Two dogs are playing by a tree.</em></td>
<td>Yes</td>
<td>Yes</td>
<td>$\forall x.\text{tree}(x) \rightarrow \text{plant}(x)$</td>
</tr>
<tr>
<td></td>
<td>H: <em>Two dogs are playing by a plant.</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2404</td>
<td>T: <em>The lady is slicing a tomato.</em></td>
<td>No</td>
<td>No</td>
<td>$\forall v.\text{slice}(v) \rightarrow \text{cut}(v)$</td>
</tr>
<tr>
<td></td>
<td>H: <em>There is no one cutting a tomato.</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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)

<table>
<thead>
<tr>
<th></th>
<th>False negatives: missing word-to-word axioms $\times 12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>530</td>
<td>T: <em>A biker is wearing gear which is black.</em></td>
</tr>
<tr>
<td></td>
<td>H: <em>A biker wearing black is breaking the gears.</em></td>
</tr>
<tr>
<td>1495</td>
<td>T: <em>A man is playing a guitar.</em></td>
</tr>
<tr>
<td></td>
<td>H: <em>A man is strumming a guitar.</em></td>
</tr>
<tr>
<td>4360</td>
<td>T: <em>A man is cutting a note.</em></td>
</tr>
<tr>
<td></td>
<td>H: <em>A man is cutting a paper.</em></td>
</tr>
</tbody>
</table>

**Others:** healing/helping, fitting/applying, ringers/wrestlers, lunging/jumping, drawing/tatoo, lady/girl, elder/elderly, woman/lady
## Introducing Lexical Knowledge

### Error analysis: WordNet

<table>
<thead>
<tr>
<th>Prob. ID</th>
<th>T-H pairs</th>
<th>Gold</th>
<th>System</th>
<th>Axioms</th>
</tr>
</thead>
<tbody>
<tr>
<td>1412</td>
<td>T: <em>Men are sawing logs</em>.</td>
<td>Yes</td>
<td>Yes</td>
<td>$\forall v.\text{saw}(v) \rightarrow \text{cut}(v)$</td>
</tr>
<tr>
<td></td>
<td>H: <em>Men are cutting wood</em>.</td>
<td></td>
<td></td>
<td>$\forall x.\text{log}(x) \rightarrow \text{wood}(x)$</td>
</tr>
<tr>
<td>211</td>
<td>T: <em>Two dogs are playing by a tree</em>.</td>
<td>Yes</td>
<td>Yes</td>
<td>$\forall x.\text{tree}(x) \rightarrow \text{plant}(x)$</td>
</tr>
<tr>
<td>2404</td>
<td>T: <em>The lady is slicing a tomato</em>.</td>
<td>No</td>
<td>No</td>
<td>$\forall v.\text{slice}(v) \rightarrow \text{cut}(v)$</td>
</tr>
<tr>
<td></td>
<td>H: <em>There is no one cutting a tomato</em>.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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*. | Unk  | Yes    | $\forall v.\text{wear}(v) \rightarrow \text{break}(v)$               |

### False negatives: missing word-to-word axioms $\times 12$

| 1495    | T: *A man is playing a guitar*.             | Yes  | Unk    | $\forall v.\text{play}(v) \rightarrow \text{strum}(v)$              |
| 4360    | T: *A man is cutting a note*.               | Yes  | Unk    | $\forall x.\text{note}(x) \rightarrow \text{paper}(x)$              |

**Others:** healing/helping, fitting/applying, ringers/wrestlers, lunging/jumping, drawing/tattoo, lady/girl, elder/elderly, woman/lady

### False negatives: missing phrase axioms $\times 38$

| 1266    | T: *A band is playing on a stage*.         | Yes  | Unk    | “on a stage” $\rightarrow$ “onstage”                                  |
| 1584    | T: *A hole is being burrowed by the badger*. | Yes  | Unk    | “burrow a hole” $\rightarrow$ “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, ...
Future work

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