Human-in-the-Loop Parsing

Luheng He, Julian Michael, *Mike Lewis, Luke Zettlemoyer

University of Washington

EMNLP 2016

*Now at Facebook AI Research
Our key hypothesis:
Anyone who **understands the meaning of a sentence** should be able to correct **parser mistakes**.
Our key hypothesis:
Anyone who understands the meaning of a sentence should be able to correct parser mistakes.

Pat ate the cake on the table that I baked last night.
Our key hypothesis:
Anyone who **understands the meaning of a sentence** should be able to correct **parser mistakes**.

Pat ate the cake on the table that I *baked* last night.
Our key hypothesis:
Anyone who **understands the meaning of a sentence** should be able to correct **parser mistakes**.

Pat ate the cake on the table that I **baked** last night.
Our key hypothesis:
Anyone who **understands the meaning of a sentence** should be able to correct **parser mistakes**.

Pat ate the cake on the table that I **baked** last night.

**Parser:** I **baked** **table**

**Human understanding:** I **baked** **cake**
Our key hypothesis:
Anyone who understands the meaning of a sentence should be able to correct parser mistakes.

Pat ate the cake on the table that I baked last night.

Parser: I baked table
Human understanding: I baked cake

How can we use this kind of human knowledge?
Pat ate the cake on the table that I *baked* last night.
Pat ate the cake on the table that I *baked* last night.
Pat ate the cake on the table that I baked last night.
Pat ate the cake on the table that I baked last night.

Q: What did someone bake?
1. table  2. cake
## Related Work

<table>
<thead>
<tr>
<th>Form of Supervision</th>
<th>Data Collection</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jha et al., 2010</td>
<td>PP Attachment Decisions</td>
<td>Crowdsourced</td>
</tr>
<tr>
<td>Choe and McClosky, 2015</td>
<td>Paraphrases</td>
<td>In-house Annotator</td>
</tr>
<tr>
<td>Duan et al., 2016</td>
<td>Paraphrases</td>
<td>Crowdsourced</td>
</tr>
</tbody>
</table>
Scope of this Work

- Target core arguments of verbal predicates.
- Use human judgments to fix parser mistakes at decoding time.
- Use CCG (Combinatory Categorial Grammar) as the underlying syntactic formalism.
- Use the Neural CCG Parser (Lewis et al. 2016) as our base parser.
Workflow

CCG Parser → Question Generator → Crowdsourcing Platform

Re-parsed CCG Dependency Tree → Re-parse w/ Constraints
Workflow

Candidate dependencies from the n-best list:
- baked → table
- baked → cake

CCG Parser

Question Generator

Re-pars ed CCG Dependency Tree

Re-parse w/ Constraints

Crowdsourcing Platform
Candidate dependencies from the n-best list:
- baked → table
- baked → cake

Q: “What did someone bake?”
1) table 2) cake

Re-parsed CCG Dependency Tree
Candidate dependencies from the n-best list:
- baked → table
- baked → cake

Q: “What did someone bake?”
1) table 2) cake

Re-parsed CCG Dependency Tree

Cake (4 votes)
Table (1 vote)

Crowdsourcing Platform
Workflow

Candidate dependencies from the n-best list:
- baked → table
- baked → cake

Q: “What did someone bake?”
1) table 2) cake

cake (4 votes)
table (1 vote)

C_pos (bake → cake)
C_neg (bake → table)
Workflow

Candidate dependencies from the n-best list:
- baked → table
- baked → cake

Q: “What did someone bake?”
1) table 2) cake

cake (4 votes)
table (1 vote)

CCG Parser

Question Generator

Crowdsourcing Platform

Re-parsed CCG Dependency Tree

Re-parse w/ Constraints

C_pos (bake → cake)
C_neg (bake → table)

Not re-training the model
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of **baked**: \((S\backslash NP_1)/NP_2\)
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of \textit{baked}: \((S\backslash NP_1)/NP_2\)

Convert to template: \[
\begin{array}{c}
\text{NP}_1 \quad \text{bake} \quad \text{NP}_2
\end{array}
\]

\[\text{I} \quad \text{baked} \quad \text{table}\]
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of baked: \((S\backslash NP_1)/NP_2\)

Convert to template: \(NP_1\) bake \(NP_2\)

Filling-in the Slots:

what bake sth.

I baked table
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of *baked*: \((S\backslash NP_1)/NP_2\)

Convert to template: \[NP_1 \text{ bake } NP_2\]

Filling-in the Slots:

What baked something?

— I
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of *baked*: \((S\backslash NP_1)/NP_2\)

Convert to template: \[ NP_1 \text{ bake } NP_2 \]

Filling-in the Slots:

**What baked something?**

— I

**What did someone bake?**

— the table
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of \textit{baked}: \((S\NP_1) / \NP_2\)

Convert to template: \[
\begin{array}{|c|c|}
\hline
\NP_1 & \text{bake} & \NP_2 \\
\hline
\end{array}
\]

Filling-in the Slots:

\[
\begin{array}{|c|c|}
\hline
\text{what} & \text{bake} & \text{sth.} \\
\hline
\end{array}
\]

\textbf{What} baked something?

\begin{itemize}
  \item I
\end{itemize}

\textbf{What} did someone bake?

\begin{itemize}
  \item the table
\end{itemize}

Infer \textbf{someone/something} and the \textbf{answer spans} based on the n-best parses

Used “\textit{what}” for all questions
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of \textit{baked}: \((S\NP_1)/\NP_2\)

Convert to template: \[\NP_1 \text{ bake } \NP_2\]

Filling-in the Slots:

- What baked something?
  - I
- What did someone bake?
  - the table

Infer \textit{someone/something} and the answer spans based on the n-best parses

Used “what” for all questions
Generate Q/A Pairs from CCG Dependencies

Predicted CCG category of **baked**: \((S\backslash NP_1)/NP_2\)

Convert to template: \[ NP_1 \text{ bake } NP_2 \]

Filling-in the Slots:

- **What** baked something?  
  - **I**

- **What** baked something?  
  - **I**

- **What** did someone bake?  
  - **the table**

- **What** did someone bake?  
  - **the cake**

Infer **someone/something** and the **answer spans** based on the n-best parses

Used “**what**” for all questions
Group Q/A Pairs into Queries

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>What baked something?</td>
<td>I</td>
</tr>
<tr>
<td>What did someone bake?</td>
<td>the table, the cake</td>
</tr>
<tr>
<td>What was baked something?</td>
<td>the table</td>
</tr>
</tbody>
</table>
### Group Q/A Pairs into Queries

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
<th>Scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>What baked something?</td>
<td>I</td>
<td>1.0</td>
</tr>
<tr>
<td>What did someone bake?</td>
<td>the table</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>the cake</td>
<td>0.3</td>
</tr>
<tr>
<td>What was baked something something?</td>
<td>the table</td>
<td>0.1</td>
</tr>
</tbody>
</table>
## Group Q/A Pairs into Queries

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
<th>Scores</th>
<th>Question Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>What baked something?</td>
<td>I</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>What did someone bake?</td>
<td>the table</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>the cake</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>What was baked something something?</td>
<td>the table</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
## Group Q/A Pairs into Queries

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
<th>Scores</th>
<th>Question Confidence</th>
<th>Answer Uncertainty (Entropy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>What baked something?</td>
<td>I</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>What did someone bake?</td>
<td>the table</td>
<td>0.7</td>
<td>1.0</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>the cake</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>What was baked something something?</td>
<td>the table</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
</tbody>
</table>
## Group Q/A Pairs into Queries

<table>
<thead>
<tr>
<th>Questions</th>
<th>Answers</th>
<th>Scores</th>
<th>Question Confidence</th>
<th>Answer Uncertainty (Entropy)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What</strong> baked something?</td>
<td>I</td>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>What</strong> did someone bake?</td>
<td>the table</td>
<td>0.7</td>
<td>1.0</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>the cake</td>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>What</strong> was baked something</td>
<td>the table</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
</tr>
<tr>
<td>something?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Non-sensical question**
- **No uncertainty**
Our Annotation Task

Sentence:
Pat ate the cake on the table that I baked last night.

Question:
What did someone bake?

Check one or more
☐ the cake
☐ the table
☐ None of the above.

Comment:

* Crowdsourcing platform: https://www.crowdflower.com/.
Data Collection with Crowdsourcing

- All developments are done on CCG-Dev only.
- Less than 2 queries per sentence, for about 60% of the sentences.
- **Cost:** 46 cents per query.
- **Speed:** 200 queries per hour.
Inter-Annotator Agreement

- Agreement is computed only for matching the exact set of answers. i.e. (A, B) and (B) are considered disagreement.
- Unanimous agreement for over 40% of the queries.
- Over 90% absolute majority.
Putting our hypothesis to the test: How well does annotators’ human understanding align with the gold syntax?

- Successes: Long-range attachment decisions
- Challenges: Syntax-semantics mismatch
- Use heuristics to fix the mismatch problems at re-parsing time.
Temple also said Sea Containers’ plan raises numerous legal, regulatory, financial and fairness issues, but didn’t *elaborate*.

What *didn’t* *elaborate* something?

Temple

Sea Containers’ plan

None of the above.
Success - Long-range Dependency

Temple also said Sea Containers’ plan raises numerous legal, regulatory, financial and fairness issues, but didn’t elaborate.

What didn’t elaborate something?

4 Temple

1 Sea Containers’ plan

0 None of the above.
Challenge - Coreference

Kalipharma is a New Jersey-based pharmaceuticals concern that **sells** products under the Purepac label.

What **sells** something?

Kalipharma

a New Jersey-based pharmaceuticals concern

None of the above.
Challenge - Coreference

Kalipharma is a New Jersey-based pharmaceuticals concern that **sells** products under the Purepac label.

What **sells** something?

- 5 Kalipharma
- 0 a New Jersey-based pharmaceuticals concern
- 0 None of the above.
Kalipharma is a New Jersey-based pharmaceuticals concern that *sells* products under the Purepac label.

What *sells* something?

5 Kalipharma
0 a New Jersey-based pharmaceuticals concern
0 None of the above.

• Syntax-semantics mismatch
• Also happens with pronouns and appositives.
• Some cases are heuristically fixed during reparsing.
Challenge - Headedness

Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications.

What would be covered?

Timex

many types of watches

duty-free treatment

watches

None of the above.
Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications.

What would be covered?

- Timex: 0
- Duty-free treatment: 0
- None of the above: 0
- Many types of watches: 2
- Watches: 3
Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications.

What would be covered?

0 Timex
0 duty-free treatment
0 None of the above.

2 many types of watches
3 watches
Challenge - Headedness

Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications.

What would be covered?

0 Timex
0 duty-free treatment
0 None of the above.

2 many types of watches
3 watches

• Annotators tend to struggle with headedness.
• We add “disjunctive constraint”, forcing the re-parser to produce either of the two dependencies.
Re-Parsing with Crowdsourced Constraints

Q1: What did someone bake?

- votes(cake) = 4
- votes(table) = 1
- votes(None of the above) = 0

\[ y^{new} = \arg \max_y \quad \text{base-parser-score}(y) \]
\[ -T^+ \times \mathbb{1}(\text{baked} \rightarrow \text{cake} \in y) \]
\[ -T^- \times \mathbb{1}(\text{baked} \rightarrow \text{table} \in y) \]

- Penalizes parses that disagree with crowdsourced judgments.
- Constraints are decomposed by dependencies.
- Thresholds and penalties are tuned on CCG-Dev.
Re-parsing Results (Labeled F1)

- Modest improvement due to syntax-semantics mismatch.
- Larger improvement on out-of-domain data.
Re-parsing Results (Labeled F1)

- Modest improvement due to syntax-semantics mismatch.
- Larger improvement on out-of-domain data.

Active, Ser133-phosphorylated CREB effects transcription of CRE-dependent genes via interaction with the 265-kDa …
Modified parse trees for about 10% of the sentences after incorporating human judgments.

Larger gain on changed sentences.

Changed sentences are “more difficult” on average.
• Modified parse trees for about 10% of the sentences after incorporating human judgments.
• Larger gain on changed sentences.
• Changed sentences are “more difficult” on average.
Future Work

• Improve coverage by adding new types of questions:
  • Modifiers: when, where, why …
  • PP attachments with natural language queries.
• Bootstrap a parser in a low-resource domain.
• Focus on downstream applications (e.g. Information Extraction).
Contributions

• Use non-expert annotation to improve a parser.

• Crowdsourced Q/A data for further exploration of active learning/reinforcement learning techniques.

• Code and data available online: https://github.com/luheng/hitl_parsing
Thank You!

CCG Parser → Question Generator

Code and Data: https://github.com/luheng/hitl_parsing

Crowdsourcing Platform

Re-parsed CCG Dependency Tree

Re-parse w/ Constraints