Human-in-the-Loop parsing

Luheng He, Julian Michael, Mike Lewis and Luke Zettlemoyer
Problem

- Large size of datasets is a bottleneck for natural language processing systems
- Proposed solution - Human-in-the-loop parsing.
- Non-experts improve parsing my answering questions automatically generated from the parser’s output

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

<table>
<thead>
<tr>
<th>Q:</th>
<th>What didn’t <em>elaborate</em>?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>Temple</td>
</tr>
<tr>
<td>[2]</td>
<td>Sea Containers’ plan</td>
</tr>
<tr>
<td>[3]</td>
<td>None of the above.</td>
</tr>
</tbody>
</table>
CCG

\[
\begin{array}{c}
\text{CCG} \\
\frac{\text{NP}}{CCG} \\
\frac{S \setminus \text{NP} \setminus \text{ADJ}}{\lambda f. \lambda x. f(x)} \\
\frac{\text{ADJ}}{\lambda x. \text{fun}(x)} \\
\frac{S \setminus \text{NP}}{\lambda x. \text{fun}(x)} \\
\frac{S}{\text{fun}(\text{CCG})}
\end{array}
\]
CCG Categories

Syntax
- Primitive symbols: NP, ADJ etc
- Syntactic combination operators (/,\)

Semantics
- λ-calculus expression

\[ ADJ : \lambda x. \text{fun}(x) \]

- Basic building block
- Capture syntactic and semantic information jointly
CCG Lexical Entries

- Pair words and phrases with meaning
- Meaning captured by a CCG category
CCG Lexicons

\[ \text{fun} \vdash ADJ : \lambda x. \text{fun}(x) \]

\[ \text{is} \vdash (S \backslash NP) / ADJ : \lambda f. \lambda x. f(x) \]

\[ \text{CCG} \vdash NP : \text{CCG} \]

- Pair words and phrases with meaning
- Meaning captured by a CCG category
CCG Operations

- Small set of operators
- Input: 1-2 CCG categories
- Output: A single CCG category
- Operate on syntax semantics together
- Mirror natural logic operations
CCG Operations

\[
\begin{align*}
A \setminus B : f &\Rightarrow A : f(g) \quad (<) \\
A / B : f &\Rightarrow A : f(g) \quad (>)
\end{align*}
\]

- Equivalent to function application
- Two directions: forward and backward
  - Determined by slash direction
CCG Parsing

Combine categories using operators
Weighted Linear CCGs

- Given a weighted linear model:
  - CCG lexicon $\Lambda$
  - Feature function $f : X \times Y \rightarrow \mathbb{R}^m$
  - Weights $w \in \mathbb{R}^m$

- The best parse is:
  $$y^* = \arg \max_y w \cdot f(x, y)$$

- We consider all possible parses $y$ for sentence $x$ given the lexicon $\Lambda$
Mapping CCG parses to queries

- Parse sentence using Combinatory Categorial Grammar (CCG) parser.
- Determine verb’s set of arguments by the CCG supertag assigned to it
- Obtain dependencies for each argument position
- Replace noun phrases by *something*

put - CCG supertag $((S\backslash NP)/(PP))/NP$
CCG supertag to dependency - “simple heuristic”

<table>
<thead>
<tr>
<th>Dependency</th>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>want → I</td>
<td>What wants to eat something?</td>
<td>I</td>
</tr>
<tr>
<td>eat → I</td>
<td>What would eat something?</td>
<td>I</td>
</tr>
<tr>
<td>eat → pizza</td>
<td>What would something eat?</td>
<td>the pizza</td>
</tr>
<tr>
<td>put → you</td>
<td>What put something?</td>
<td>you</td>
</tr>
<tr>
<td>put → pizza</td>
<td>What did something put?</td>
<td>the pizza</td>
</tr>
<tr>
<td>on → table</td>
<td>What did something put something on?</td>
<td>the table</td>
</tr>
</tbody>
</table>
Mapping CCG parses to queries

- Generate Q for every parse in 100-best outputs of the parser
- Pool Q by the head of the dependency, its CCG category and question string
- Each pool becomes a query
- Compute marginalized score for each QA phrase by summing over scores of all parses that generated them
- For each unique dependency, add candidate answer to the query by choosing the answer phrase that has the highest marginalized score for that dependency
- Remove queries and answers with marginalized score below certain threshold, and queries with one answer (only keep confident questions with uncertain answers)
## Examples

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Question</th>
<th>Votes</th>
<th>Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Structural Dynamics Research Corp. . . . said it introduced new technology in mechanical design automation that will improve mechanical engineering productivity.</td>
<td>What will improve something?</td>
<td>0</td>
<td>structural dynamics research corp new technology mechanical design automation</td>
</tr>
<tr>
<td>(2) He said disciplinary proceedings are confidential and declined to comment on whether any are being held against Mr. Trudeau.</td>
<td>What would comment?</td>
<td>5</td>
<td>he disciplinary proceedings</td>
</tr>
<tr>
<td>(3) To avoid these costs, and a possible default, immediate action is imperative.</td>
<td>What would something avoid?</td>
<td>4</td>
<td>these costs a possible default</td>
</tr>
<tr>
<td>(4) The price is a new high for California Cabernet Sauvignon, but it is not the highest.</td>
<td>What is not the highest?</td>
<td>2</td>
<td>the price</td>
</tr>
<tr>
<td>(5) Kalipharma is a New Jersey-based pharmaceuticals concern that sells products under the Purepac label.</td>
<td>What sells something?</td>
<td>5</td>
<td>kalipharma a new jersey-based pharmaceuticals concern</td>
</tr>
<tr>
<td>(6) Further, he said, the company doesn’t have the capital needed to build the business over the next year or two.</td>
<td>What would build something?</td>
<td>4</td>
<td>the company the capital</td>
</tr>
<tr>
<td>(7) Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications.</td>
<td>What would be covered?</td>
<td>0</td>
<td>timex duty-free treatment</td>
</tr>
<tr>
<td>(8) You either believe Seymour can do it again or you don’t.</td>
<td>What does?</td>
<td>3</td>
<td>you some of the above</td>
</tr>
</tbody>
</table>

Table 2: Example annotations from the CCGbank development set. Answers that agree with the gold parse are in bold. The answer choice None of the above was present for all examples, but we only show it when it was chosen by annotators.
Re-parsing with QA annotation

- For question $q$, with answer $a$, denote by $v(a)$ the fraction/number of annotators that chose $a$.
- Add re-parsing constraints as follows

  - If $v(\text{None of the above}) \geq T^+$, penalize parses that agree with $q$’s supertag on the verb by $w^t$
  - If $v(a) \leq T^-$, penalize parses containing $d$ by $w^-$
  - If $v(a) \geq T^+$, penalize parses that do not contain $d$ by $w^+$

<table>
<thead>
<tr>
<th>Data</th>
<th>L16</th>
<th>HITL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCG-Dev</td>
<td>87.9</td>
<td>88.4</td>
</tr>
<tr>
<td>CCG-Test</td>
<td>88.1</td>
<td>88.3</td>
</tr>
<tr>
<td>Bioinfer</td>
<td>82.2</td>
<td>82.8</td>
</tr>
</tbody>
</table>

Table 6: CCG parsing accuracy with human in the loop (HITL) versus the state-of-the-art baseline (L16) in terms of labeled F1 score. For both in-domain and out-domain, we have a modest gain over the entire corpus.

<table>
<thead>
<tr>
<th>Data</th>
<th>L16</th>
<th>HITL</th>
<th>Pct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCG-Dev</td>
<td>83.9</td>
<td>87.1</td>
<td>12%</td>
</tr>
<tr>
<td>CCG-Test</td>
<td>84.2</td>
<td>85.9</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 7: Improvements of CCG parsing accuracy on changed sentences for in-domain data. We achieved significant improvement over the 10%–12% (Pct.) sentences that were changed by re-parsing.