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***.	
Q:	What didn’t ***elaborate***?
[1] ****	Temple
[2] *	Sea Containers’ plan
[3]	None of the above.

CCG

$$
\begin{array}{ccc}
\text{CCG} & \text{is} & \text{fun} \\
\hline
NP & S\backslash NP/ADJ & ADJ \\
CCG & \lambda f.\lambda x.f(x) & \lambda x.fun(x) \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP \quad >} \\
 & \multicolumn{2}{c}{\lambda x.fun(x)} \\
\hline
\multicolumn{3}{c}{S \quad <} \\
\multicolumn{3}{c}{fun(CCG)}
\end{array}
$$

CCG Categories

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

$$ADJ : \lambda x.fun(x)$$

Semantics
-λ-calculus expression

- Basic building block
- Capture syntactic and semantic information jointly

CCG Lexical Entries

$$\underbrace{\text{fun}}_{\text{Natural Language}} \vdash \underbrace{ADJ : \lambda x.fun(x)}_{\text{CCG Category}}$$

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

CCG Lexicons

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

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

$$\text{CCG} \vdash NP : CCG$$

- 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

$$\overset{\text{Argument}}{\boxed{B : g}} \quad \overset{\text{Function}}{\boxed{A \backslash B : f}} \Rightarrow \overset{\text{Result}}{\boxed{A : f(g)}} \qquad (<)$$

$$A/B : f \quad B : g \Rightarrow A : f(g) \qquad (>)$$

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

CCG Parsing

$$
\begin{array}{ccc}
\text{CCG} & \text{is} & \text{fun} \\
\hline
NP & S\backslash NP/ADJ & ADJ \\
CCG & \lambda f.\lambda x.f(x) & \lambda x.fun(x) \\
\hline
 & \multicolumn{2}{c}{S\backslash NP \quad >} \\
 & \multicolumn{2}{c}{\lambda x.fun(x)}
\end{array}
$$

Combine categories using operators

Weighted Linear CCGs

- Given a weighted linear model:
  - CCG lexicon Λ
  - Feature function $f : X \times Y \to \mathbb{R}^m$
  - Weights $w \in \mathbb{R}^m$
- The best parse is:

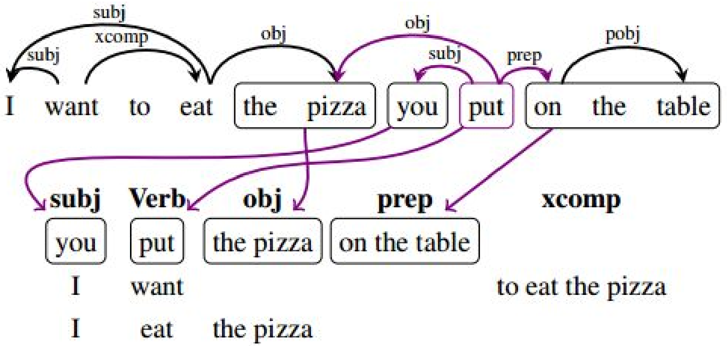
$$y^* = \arg\max_y w \cdot f(x, y)$$

- We consider all possible parses y for sentence x given the lexicon Λ

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"



Dependency	Question	Answer
want → I	What wants to eat something?	I
eat → I	What would eat something?	I
eat → pizza	What would something eat?	the pizza
put → you	What put something?	you
put → pizza	What did something put?	the pizza
on → table	What did something put something on?	the 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, it’s 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

Sentence	Question	Votes	Answers
(1) Structural Dynamics Research Corp. . . . said it introduced new technology in mechanical design automation that will ***improve*** mechanical engineering productivity.	What will ***improve*** something?	0 5 0	Structural Dynamics Research Corp **new technology** mechanical design automation
(2) He said disciplinary proceedings are confidential and declined to ***comment*** on whether any are being held against Mr. Trudeau.	What would ***comment***?	5 0	**he** disciplinary proceedings
(3) To ***avoid*** these costs, and a possible default, immediate action is imperative.	What would something ***avoid***?	4 3	**these costs** **a possible default**
(4) The price is a new high for California Cabernet Sauvignon, but it ***is*** not the highest.	What ***is*** not the highest?	2 3	the price **it**
(5) Kalipharma is a New Jersey-based pharmaceuticals concern that ***sells*** products under the Purepac label.	What ***sells*** something?	5 0	Kalipharma **a New Jersey-based pharmaceuticals concern**
(6) Further, he said, the company doesn't have the capital needed to ***build*** the business over the next year or two.	What would ***build*** something?	4 1	the company **the capital**
(7) Timex had requested duty-free treatment for many types of watches, ***covered*** by 58 different U.S. tariff classifications.	What would be ***covered***?	0 0 2 3	Timex duty-free treatment **many types of watches** watches
(8) You either believe Seymour can do it again or you ***do*** n't .	What ***does***?	3 0 2	**you** Seymour None of the above

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 $v(\textit{None of the above}) \geq T^+$, 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^+

Data	**L16**	**HITL**
CCG-Dev	87.9	88.4
CCG-Test	88.1	88.3
Bioinfer	82.2	82.8

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.

Data	**L16**	**HITL**	**Pct.**
CCG-Dev	83.9	87.1	12%
CCG-Test	84.2	85.9	10%

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.