Linear-time Parsing Using Combinatory Categorial Grammar (CCG)

Wenduan Xu, Yue Zhang and Stephen Clark, Natural Language and Information Processing Group

Background

Task: interpreting the syntactic and semantic structures of English sentences.
Usage: a parser is a basic component for:
- Web search (Google)
- Automatic translation (Google Translate)
- Question answering (Siri)
- Almost all language technologies

Challenge: Ambiguity

The following is a parser tree of a typical newspaper sentence.

Canadian Utilities had 1988 revenue of C$ 1.16 billion, mainly from its natural gas and electric utility businesses in Alberta, where the company serves about 800,000 customers.

The number of possible analyses grows exponentially (as the Catalan number) with respect to sentence length and brute-force exhaustive-search is prohibitively slow.

Semantic Ambiguity

Different syntactic structures lead to different semantic interpretations.

<table>
<thead>
<tr>
<th>VP</th>
<th>NP</th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>saw</td>
<td>DT</td>
<td>NN</td>
</tr>
<tr>
<td>put</td>
<td>DT</td>
<td>NN</td>
</tr>
<tr>
<td>a</td>
<td>DT</td>
<td>NN</td>
</tr>
<tr>
<td>telescope</td>
<td>DT</td>
<td>NN</td>
</tr>
</tbody>
</table>

In the above two parses, “with a telescope” is either attached to “saw” or to “the man”.

Probabilistic Parsing

A parser tries to find the most probable analysis of a given sentence, according to a probabilistic disambiguation model.

Better disambiguation leads to better parsing accuracy.

Why use CCG?

Long-range Dependency

Mark Steedman’s Home Page

Address:
School of Informatics
University of Edinburgh
Informatics Forum 415
10 Crichton Street
Edinburgh, EH8 9AB
Scotland, United Kingdom

Email: If you aren’t a robot, and/or you are equipped with a parser that can handle long-range dependencies, you will be able to email me at an address formed by concatenating my surname, the “at” thing, the first three letters of the thing that the address above says I’m in the school of, and the string dot ed dot ac dot uk

Tel: +44 (131) 650 4631
Fax: +44 (131) 650 6626

Parsers need to resolve long-range dependencies such as those contained in the above email statement in the highlighted box.

Human-like Parsing

In this work, we use CCG to achieve human-like sentence processing.

- We parse by reading from left to right, and resolve ambiguity incrementally as we read.
- We interpret both syntactic (grammar) and semantic (meaning) information on-the-fly simultaneously.
- We dump the syntax once a sentence is parsed, producing semantic meaning as output.

CCG is a mildly context-sensitive grammar formalism that is well-suited to capture many advanced linguistic phenomena, including long-range dependencies.

CCG = Incremental left-to-right analysis + Integrated syntactic & semantic interpretation + Output "semantics" as parsing output

After parsing the above sentence, our CCG parser will output the subject is the boy, the thing being eaten is the cake and the action is eat.

Left-to-right Shift-Reduce Parsing

Anatomy of the Parser

Parser

Inference Algorithm

Training Data

Parser

CCG Parsing Model

Probabilistic Inference Algorithm

Shift-Reduce Decoding

Input: IBM (NP) bought (S/NP/NP) Lotus (NP)

- Before parsing starts, each input word is assigned a CCG category. A CCG category represents how a word interacts with other words.
- Both IBM and Lotus are noun-phrases (NP): bought is a transitive verb.
- The CCG category (S/NP/NP) for a transitive verb means it first takes an object to its right (the NP after the forward slash) and it takes a subject to its left (the NP after the backward slash), producing a sentence, represented by the S category.

Shift-Reduce Decoding

P (reduce) = ?

IBM

stack

bought

Lotus

P (Shift) = ?

queue

The parser makes probabilistic decisions to either read in (shift) one or more words each time, or generate semantic interpretations (reduce) from those words already read.

- We expect the parser to output, for example, the subject of bought is IBM and the object of bought is Lotus.
- Our parser is the best-performing shift-reduce parser for CCG to date, achieving 85.96% labelled dependency F-score for the standard Wall Street Journal test data.