Introduction to Syntax and Parsing
ACS 2015/16
Stephen Clark
L7: A CCG Grammar and Treebank for naturally occurring text
Pierre Vinken, 61 years old, will join the board as a non-executive director Nov. 29.

Activation of the CD28 surface receptor provides a major costimulatory signal for T cell activation resulting in enhanced production of interleukin-2 (IL-2) and cell proliferation.

The Trust’s symbol, a sprig of oak leaves and acorns, is thought to have been inspired by a carving in the cornice of the Alfriston Clergy House.

• Can we really move from simple “linguistic” examples to sentences like these found in the real world?
Newspaper Example

Pierre Vinken, 61 years old, will join the board as a nonexecutive director on Nov. 29.

- Needs an $N \rightarrow NP$ rule
- $S[adj]\ NP$ is for predicative adjectives, e.g. the man is old
- We need a unary type-changing rule: $S[adj]\ NP \rightarrow NP/NP$
- We need special rules in the parser to deal with punctuation
- Only need application in this example (no composition or type-raising)
Grammatical Features in CCGBank

- $S$ category often has a grammatical feature which indicates the kind of sentence or verb phrase
  - $S[dcl]$ declarative sentence
  - $S[q]$ yes/no questions
  - $S[b]$ bare infinitives
  - $S[to]$ to infinitives
  - $S[pss]$ past participles in passive mode
  - $S[pt]$ past participles in active mode
  - $S[ng]$ present participles
  - ... 

- See p.47 of Julia’s thesis for full list

- $S$ in adverbial modifiers, e.g. $(S\backslash NP)/(S\backslash NP)$, effectively has a variable feature: $(S[X]\backslash NP)/(S[X]\backslash NP)$, which unifies with the feature on the argument and transfers to the result
Activation of (NP\NP)/NP the NP/N CD28/N/N surface/N/N receptor/N provides (S[dcl]\NP)/NP a NP/N major N/N costimulatory N/N signal N for (NP\NP)/NP T|NP N/N cell N/N activation N resulting (S[ng]\NP)/PP in PP/NP enhanced N/N production N of (NP\NP)/NP interleukin-2 N ((IL-2|N )) and conj cell N/N proliferation N.

- Needs a unary type-changing rule: $S[ng]\NP \rightarrow (S\NP)\(S\NP)$
- Need special rules to deal with brackets
- Still only needs application
Wikipedia Example

The Trust's symbol, a sprig of oak leaves and acorns, is thought to have been inspired by a carving in the cornice of Alfriston Clergy House.

- Still only need application
- No unary type-changing rules in this example
Unary Type-Changing Rules

- Without type-changing rules (notice that the category for *used* is non-standard and the category for *once* changes also):

  \[
  \begin{array}{l}
  \text{A form of asbestos} \quad \text{once} \quad \text{used} \quad \text{to make Kent cigarettes} \\
  NP \quad \frac{(NP\backslash NP)/(NP\backslash NP)}{(NP\backslash NP)/(S[to]\backslash NP)} \quad S[to]\backslash NP
  \end{array}
  \]

- With type-changing rules (uses standard categories for *used* and *once*):

  \[
  \begin{array}{l}
  \text{A form of asbestos} \quad \text{once} \quad \text{used} \quad \text{to make Kent cigarettes} \\
  NP \quad \frac{(S\backslash NP)/(S\backslash NP)}{(S[pss]\backslash NP)/(S[to]\backslash NP)} \quad S[to]\backslash NP \\
  \quad \frac{S[pss]\backslash NP}{NP\backslash NP}
  \end{array}
  \]

- Type-changing rules increase the compactness of the lexicon (capturing generalisations) and reduce the number of categories assigned to modifiers such as *once*
Real Examples using Composition

- Object extraction from a relative clause, using type-raising and forward composition:

  \[
  \begin{array}{c|c|c|c|c}
  \text{That} & \text{finished} & \text{the job} & \text{that} & \text{Captain Chandler} \\
  \text{NP} & (S[dcl]/\text{NP})/\text{NP} & \text{NP} & (NP/\text{NP})/(S[dcl]/\text{NP}) & \text{NP} \\
  \end{array}
  \]

  \[
  \begin{array}{c|c|c}
  \text{had} & \text{begun} \\
  (S[dcl]/\text{NP})/(S[pt]/\text{NP}) & (S[pt]/\text{NP})/\text{NP} \\
  \end{array}
  \]

- Question with an object extraction:

  \[
  \begin{array}{c|c|c|c|c|c}
  \text{What} & \text{books} & \text{did} & \text{he} & \text{author} \\
  (S[wq]/(S[q]/\text{NP}))/\text{N} & \text{NP} & (S[q]/(S[b]/\text{NP}))/\text{NP} & \text{NP} & (S[b]/\text{NP})/\text{NP} \\
  \end{array}
  \]

Lots more real CCG data on my RESOURCES webpage
Creating a Treebank for CCG

- A CCG treebank consists of (sentence, CCG analysis) pairs
- The CCG analysis is likely to be a derivation, and may also contain additional information such as predicate-argument dependencies
- The treebank is useful for:
  - deriving a wide-coverage grammar (or extending an existing one)
  - inducing statistical disambiguation models
- How can we build a CCG treebank?
  - manually from scratch (or at least by correcting the output of an existing CCG parser)
  - by automatically transforming the analyses from an existing treebank (e.g. the Penn Treebank) into CCG derivations
- Manual creation of a treebank is expensive so we choose the 2nd option
The Penn Treebank

- 50k sentences/1M words of WSJ text annotated with phrase-structure (PS) trees
- How might we turn this into a CCG treebank?
- What information do we need in the PS trees?
  - head information
  - argument/adjunct distinction (so we can derive the CCG categories)
  - trace information/extracted arguments so we can analyse long-range dependencies
Example PTB Tree (with traces)
The Basic Translation Algorithm

- Ignoring long-range dependency/trace information, the basic algorithm is straightforward:
  
  - foreach tree $\tau$
    
    * determineConstituentTypes($\tau$)
    * makeBinary($\tau$)
    * assignCategories($\tau$)
Determining Constituent Type

- Constituent type is either head, complement or adjunct
- This information is not marked explicitly in the PTB, but can be inferred (using heuristic rules) based on:
  - function tags in the PTB, e.g. –SBJ (subject), –TMP (temporal modifier), –DIR (direction)
  - constituent label of a node and its parent (e.g. NP daughters of VPs are complements, unless they carry a function tag such as –LOC, –DIR, –TMP and so on)
- Appendix A of Collins’ thesis gives a list of the head rules
- See p.362 of H&S 2007 and Appendix A of CCGbank manual
Binarizing the Tree

- A PTB tree is not binarized, whereas a CCG derivation is.
- Insert dummy nodes into the tree such that:
  - all children to the left of the head branch off in a right-branching tree
  - all children to the right of the head branch off in a left-branching tree
- Some PTB structures are very flat, e.g. compound noun phrases – in the compound noun case we just assume a right-branching structure (but see Vadas and Curran for inserting NP structure into the PTB)
- See p.362 of H&S 2007
Assigning Categories

• The root node
  – mapping from categories of root nodes of PTB trees to CCG categories, e.g. \( \{VP\} \rightarrow S\backslash NP, \{S, SINV, SQ\} \rightarrow S \)

• Head and complement
  – category of complement child defined by a similar mapping, e.g. \( \{NP\} \rightarrow NP, \{PP\} \rightarrow PP \)
  – category of the head is a function which takes the category of the complement as argument and returns the category of the parent node; direction of the slash is given by the position of the complement relative to the head

• Head and adjunct
  – given a parent category \( C \), the category of an adjunct child is \( C / C \) if the adjunct child is to the left of the head child (a premodiﬁer), or \( C \backslash C \) if it is to the right (postmodiﬁer)
Long-Range Dependencies

(NP-SBJ (NP Brooks Brothers))

(SBAR (WHNP-1 (WDT which))

(S (NP-SBJ NNP Marks))

(VP (VBD bought)

(NP (–NONE– *T*-1))

(NP–TMP last year))))

• The co-indexed trace element *T*-1 is crucial in assigning the correct categories
  – used as an indication of the presence of a direct object for the verb
  – used to assign the correct category to the Wh-pronoun (using a similar mechanism to GPSG’s “slash-passing”)

• p.57 of the CCGbank manual has a detailed example
Properties of CCGbank

- 99.4% of the sentences in the PTB are translated into CCG derivations
- Words with the most number of category types:

<table>
<thead>
<tr>
<th>Word</th>
<th>num cats</th>
<th>Freq</th>
<th>Word</th>
<th>num cats</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>as</td>
<td>130</td>
<td>4237</td>
<td>of</td>
<td>59</td>
<td>22782</td>
</tr>
<tr>
<td>is</td>
<td>109</td>
<td>6893</td>
<td>that</td>
<td>55</td>
<td>7951</td>
</tr>
<tr>
<td>to</td>
<td>98</td>
<td>22056</td>
<td>LRB</td>
<td>52</td>
<td>1140</td>
</tr>
<tr>
<td>than</td>
<td>90</td>
<td>1600</td>
<td>not</td>
<td>50</td>
<td>1288</td>
</tr>
<tr>
<td>in</td>
<td>79</td>
<td>15085</td>
<td>are</td>
<td>48</td>
<td>3662</td>
</tr>
<tr>
<td>–</td>
<td>67</td>
<td>2001</td>
<td>with</td>
<td>47</td>
<td>4214</td>
</tr>
<tr>
<td>’s</td>
<td>67</td>
<td>9249</td>
<td>so</td>
<td>47</td>
<td>620</td>
</tr>
<tr>
<td>for</td>
<td>66</td>
<td>7912</td>
<td>if</td>
<td>47</td>
<td>808</td>
</tr>
<tr>
<td>at</td>
<td>63</td>
<td>4313</td>
<td>on</td>
<td>46</td>
<td>5112</td>
</tr>
<tr>
<td>was</td>
<td>61</td>
<td>3875</td>
<td>from</td>
<td>46</td>
<td>4437</td>
</tr>
</tbody>
</table>
More Statistics

- Lexicon has 74,669 entries for 44,210 word types (929,552 tokens)
- Average number of lexical categories per *token* is 19.2
- 1,286 lexical category types in total
  - 439 categories occur only once
  - 556 categories occur 5 times or more
- Coverage on unseen data: lexicon contains correct categories for 94% of tokens in section 00
  - 3.8% due to unknown words
  - 2.2% known words but not with the relevant category