Introduction to Syntax and Parsing
ACS 2015/16
Stephen Clark
L7: A CCG Grammar and Treebank for naturally occurring text

UNIVERSITY OF CAMBRIDGE

CCG Analyses for Real 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?

UNIVERSITY OF CAMBRIDGE

Newspaper Example

```
Pierre|N/N Vinken|N ,|, 61|N/N years|N old|(S[adj]\NP)\NP
,|, will|(S[dcl]\NP)/(S[b]\NP) join|((S[b]\NP)/PP)/NP
the|NP/N board|N as|PP/NP a|NP/N nonexecutive|N/N
director|N Nov.|((S\NP)\(S\NP))/N 29|N .|.
```

- Needs an $N \rightarrow NP$ rule
- $S[adj]\backslash NP$ is for predicative adjectives, e.g. *the man is old*
- We need a *unary type-changing rule*: $S[adj]\backslash NP \rightarrow NP\backslash NP$
- 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

UNIVERSITY OF CAMBRIDGE

Biomedical Example

```
Activation|N 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|(N/N)/(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: $S[ng]\backslash NP \rightarrow (S\backslash NP)\backslash(S\backslash NP)$
- Need special rules to deal with brackets
- Still only needs application

UNIVERSITY OF CAMBRIDGE

Wikipedia Example

```
The|NP/N Trust|N 's|(NP/N)\NP symbol|N ,|, a|NP/N sprig|N
of|(NP\NP)/NP oak|N/N leaves|N and|conj acorns|N ,|,
is|(S[dcl]\NP)/(S[pss]\NP) thought|(S[pss]\NP)/(S[to]\NP)
to|(S[to]\NP)/(S[b]\NP) have|(S[b]\NP)/(S[pt]\NP)
been|(S[pt]\NP)/(S[pss]\NP) inspired|S[pss]\NP
by|((S\NP)\(S\NP))/NP a|NP/N carving|N in|(NP\NP)/NP
the|NP/N cornice|N of|(NP\NP)/NP the|NP/N
Alfriston|(N/N)/(N/N) Clergy|N/N House|N .|.
```

- 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):

$$\begin{array}{cccc}
\mathit{A\ form\ of\ asbestos} & \mathit{once} & \mathit{used} & \mathit{to\ make\ Kent\ cigarettes} \\
\hline
NP & (NP\backslash NP)/(NP\backslash NP) & (NP\backslash NP)/(S[to]\backslash NP) & S[to]\backslash NP
\end{array}$$

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

$$\begin{array}{cccc}
\mathit{A\ form\ of\ asbestos} & \mathit{once} & \mathit{used} & \mathit{to\ make\ Kent\ cigarettes} \\
\hline
NP & (S\backslash NP)/(S\backslash NP) & (S[pss]\backslash NP)/(S[to]\backslash NP) & S[to]\backslash NP \\
\cline{2-4}
 & \multicolumn{3}{c}{S[pss]\backslash NP} \\
\cline{2-4}
 & \multicolumn{3}{c}{NP\backslash NP}
\end{array}$$

- 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:

That	*finished*	*the job*	*that*	*Captain Chandler*	*had*	*begun*
NP	$(S[dcl]\backslash NP)/NP$	NP	$(NP\backslash NP)/(S[dcl]/NP)$	NP	$(S[dcl]\backslash NP)/(S[pt]\backslash NP)$	$(S[pt]\backslash NP)/NP$

- Question with an object extraction:

What	*books*	*did*	*he*	*author*	?
$(S[wq]/(S[q]/NP))/N$	N	$(S[q]/(S[b]\backslash NP))/NP$	NP	$(S[b]\backslash NP)/NP$	.

Lots more real CCG data on my RESOURCES webpage

UNIVERSITY OF CAMBRIDGE

Creating a Treebank for CCG

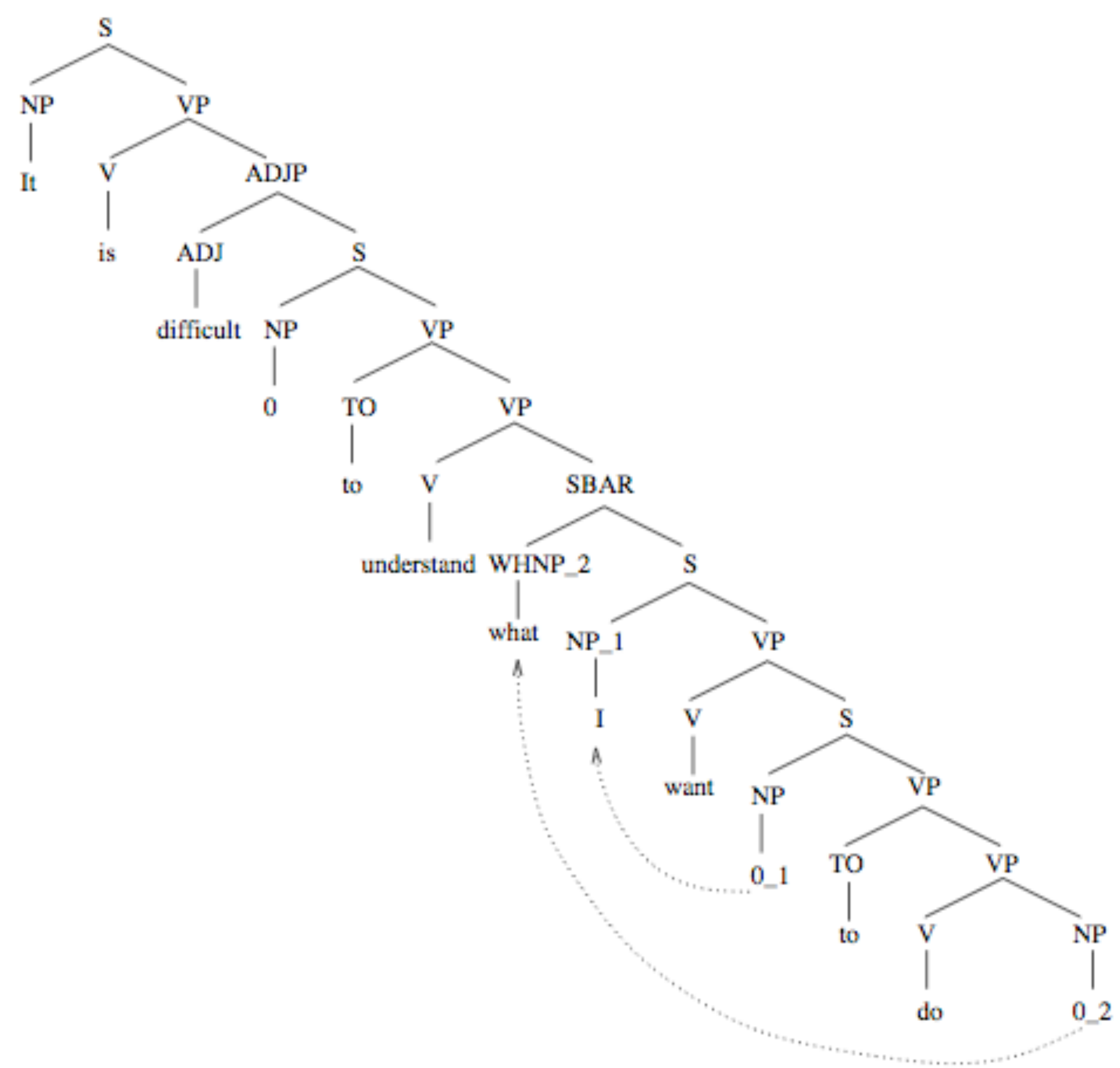
- A CCG treebank consists of (sentence, CCG analysis) pairs
- The CCG analysis is likely to be a derivation, and may also contain additional information such as predicate-argument dependencies
- The treebank is useful for:
  - deriving a wide-coverage grammar (or extending an existing one)
  - inducing 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

UNIVERSITY OF CAMBRIDGE

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)



UNIVERSITY OF CAMBRIDGE

The Basic Translation Algorithm

- Ignoring long-range dependency/trace information, the basic algorithm is straightforward:
  - foreach tree τ
    - determineConstituentTypes(τ)
    - makeBinary(τ)
    - assignCategories(τ)

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

UNIVERSITY OF CAMBRIDGE

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:

Word	num cats	Freq	Word	num cats	Freq
as	130	4237	*of*	59	22782
is	109	6893	*that*	55	7951
to	98	22056	*LRB*	52	1140
than	90	1600	*not*	50	1288
in	79	15085	*are*	48	3662
—	67	2001	*with*	47	4214
's	67	9249	*so*	47	620
for	66	7912	*if*	47	808
at	63	4313	*on*	46	5112
was	61	3875	*from*	46	4437

UNIVERSITY OF CAMBRIDGE

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 uneen 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