ACS Syntax and Semantics of Natural Language

Lecture 1: Introduction to Categorial Grammar

UNIVERSITY OF CAMBRIDGE

Stephen Clark

Natural Language and Information Processing (NLIP) Group

sc609@cam.ac.uk

Categorial Grammar (CG)

- Main responsibility for defining syntactic form is in the **lexicon**
- Hence CG is a **lexicalized** theory of grammar
  – along with other theories of grammar such as HPSG, TAG, LFG, ...
- Attractive linguistically because all language-dependent properties reside in the lexicon
  – small number of combination rules are language-invariant
- Also attractive computationally; e.g. *supertagging* for Categorial Grammar leads to highly efficient parsing (Clark and Curran, 2007)

Connection with Semantics

- Categorial Grammar has a strong commitment to Frege's **Principle of Compositionality** (along with Montague from the 70s):
- *The meaning of a phrase is a function of the meanings of the parts of the phrase and how those parts are put together*
  - mathematically often described as a *homomorphism* (structure-preserving mapping) between syntactic and semantic representations (more on this later . . .)

Contrast with Phrase-Structure Rules

- Early Chomskian approach and much work in Generative Grammar uses rewrite rules or productions (as in a Context Free Grammar):

$$S \rightarrow NP \;\; VP$$
$$VP \rightarrow TV \;\; NP$$
$$TV \rightarrow \{likes, sees, \ldots\}$$

- Categorial Grammar captures the same information by assigning a functional type, or *category*, to grammatical entities
- Has roots in early work by Polish mathematician Ajdukiewicz (1935) and even earlier in Russel's theory of types

Lexical Categories

- An elementary syntactic structure – a **lexical category** – is assigned to each word in a sentence, eg:

  walked: S\NP ‘give me an NP to my left and I return a sentence’

- Think of the lexical category for a verb as a *function*: NP is the argument, S the result, and the slash indicates the direction of the argument

Lexical Categories

- Atomic categories: S, N, NP, PP, $\ldots$ (not many more)
- Complex categories are built recursively from atomic categories and slashes
- Example complex categories for verbs:
  - intransitive verb: $S\backslash NP$ *walked*
  - transitive verb: $(S\backslash NP)/NP$ *respected*
  - ditransitive verb: $((S\backslash NP)/NP)/NP$ *gave*

A Simple CG Derivation

$$\dfrac{interleukin-10}{NP} \quad \dfrac{inhibits}{(S\backslash NP)/NP} \quad \dfrac{production}{NP}$$

A Simple CG Derivation

$$
\cfrac{\cfrac{interleukin-10}{NP} \quad \cfrac{inhibits}{(S\backslash NP)/NP} \quad \cfrac{production}{NP}}{}
$$

$$
\begin{array}{ccc}
\underline{\textit{interleukin} - \textit{10}} & \underline{\textit{inhibits}} & \underline{\textit{production}} \\
NP & (S\backslash NP)/\color{blue}{NP} & \color{blue}{NP} \\
 & \multicolumn{2}{c}{\overline{}{>}} \\
 & \multicolumn{2}{c}{S\backslash NP}
\end{array}
$$

$>$ forward application

A Simple CG Derivation

$$
\begin{array}{ccc}
\textit{interleukin} - \textit{10} & \textit{inhibits} & \textit{production} \\
\hline
NP & (S\backslash NP)/NP & NP \\
\cline{2-3}
 & \multicolumn{2}{c}{S\backslash NP} \quad > \\
\hline
\multicolumn{3}{c}{S} \quad <
\end{array}
$$

$>$ forward application

$<$ backward application

Combination Rules in CG

- Can think of the categories in blue as “cancelling”
  - early work in CG talks about “cancellation rules”
- Also looks a bit like multiplication and division
- But fundamentally the lexical category for the verb is a function which is applied to its argument

A Simple CG Derivation

- ‘Classical’ Categorial Grammar only has application rules
- Classical Categorial Grammar is context free
- So what is different to CFG?
  – lexicalisation means that the information in CFG rewrite rules has been pushed down to the leaves of the derivation

[show graphic]

Readings

- Categorial Grammar, Mark Steedman, 1999. Short encyclopedia entry for MIT Encyclopedia of Cognitive Sciences, R. Wilson and F. Keil (eds.) available at: http://homepages.inf.ed.ac.uk/steedman/papers.html
- Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models, Stephen Clark and James R. Curran, Computational Linguistics, 33(4), pp.493-552, 2007 (available online)
- My video lecture on statistical parsing with CCG (pointer from my web page): http://videolectures.net/clspss09_clark_lspl/