An Introduction to Categorial Grammar and its uses in Natural Language Processing

Gopu R Unnithan, Ashwin T V

Department of Computer Science and Automation
Indian Institute of Science

1-Dec-2015

Introduction

- Lexicalized grammar formalism
- Phrase Structured Grammar.
- Constituents and NOT dependents
- Based on the works of Kazimierz Ajdukiewicz and Yehoshua Bar-Hillel
- Combinatory Categorial Grammar(CCG) - Pioneered by Steedman and Szabolcsi.(More on this later)

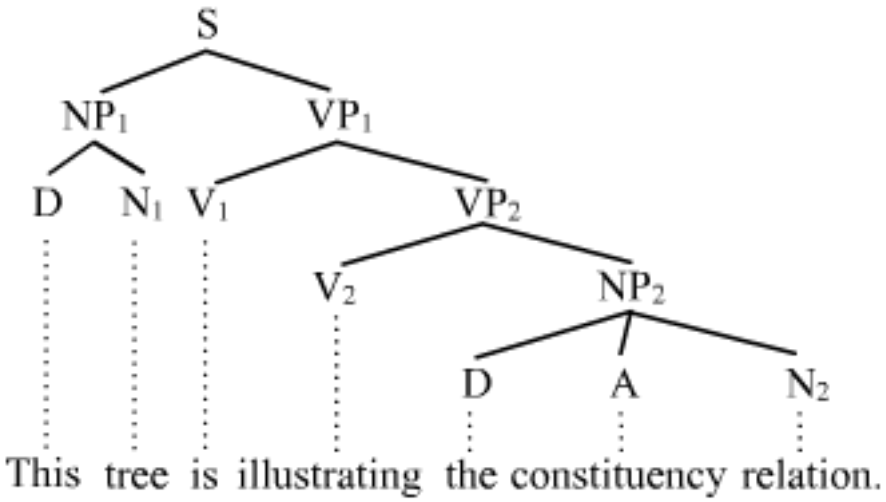
Phrase Structure Grammar

- Defined by Phrase Structure rules or rewrite rules
- $A \rightarrow B \quad C$ where A is constituent, B and C are subconstituents

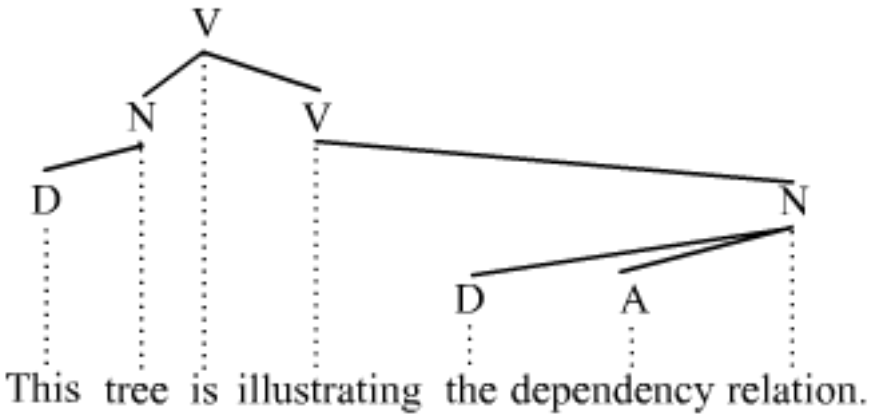
$$
\begin{aligned}
S \rightarrow & \quad NP \quad VP \\
P \rightarrow & \quad (Det) \quad N1 \\
N1 \rightarrow & \quad (AP) \quad N1 \quad (PP)
\end{aligned}
$$

- Also known as Constituency grammar
- Main trait: adherence to the constituency relation

Phrase Structure Grammar



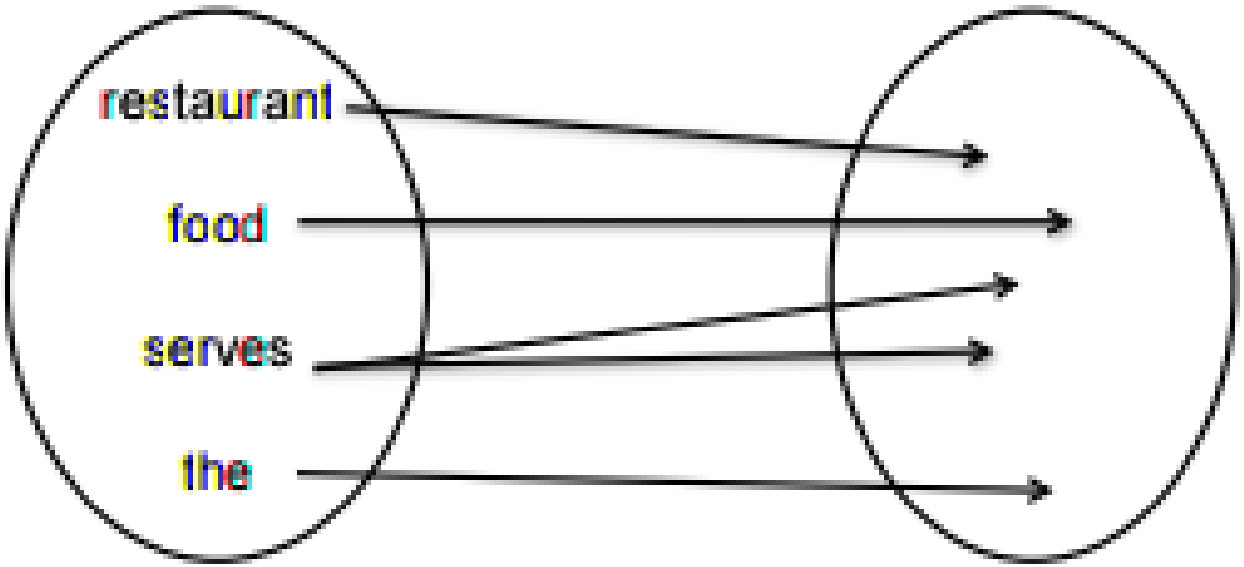
Constituency relation (PSG)



Dependency relation

Categorial Grammar

- A CG is a mapping from words to categories
- A set of word-category pairs
- What do categories look like?



Mapping not one-to-one!

Category

Two kinds of category

- **atomic** categories
- **Complex** categories

Atomic Categories

Each CG is built around a finite set of atomic categories

- simple, non-composite, atomic symbols
- similar to the symbols of a CFG

Examples:

- S - sentence/clause
- NP - noun phrase
- N - noun
- PP - preposition phrase

Complex Categories

Complex categories are built up from atomic category symbols

- From any finite set of atomic categories, can construct an infinite set of complex categories using two operators
- directional slash operators: / and \

Example

Category	**Meaning**
(S\NP)	verb phrase, intransitive verb
(NP/N)	determiner
(N/N)	adjective
(PP/NP)	preposition
((S\NP)/NP)	transitive verb

Application

What does X/Y mean?[Forward Application]
The kind of word or phrase that combines with a following Y to form an X.

$$\frac{X/Y \qquad Y}{X}\rightarrow$$

What does X\Y mean?[Backward Application]
The kind of word or phrase that combines with a preceding Y to form an X.

$$\frac{Y \qquad X\backslash Y}{X}\leftarrow$$

Categorial Grammar

- Principle of compositionality
- Grammatical constituents are distinguished by a syntactic type identifying them as either a function from arguments of one type to results of another, or as an argument(e.g. (S\NP)/NP).
- Categorial Grammar consists of lexicons and type inference rules
- Similiar to Simply typed lambda calculus.
- Advantage is that the type inference rules can be fixed permanently, so that the specification of a particular language grammar is entirely determined by the lexicon

Categorial Grammar

Definition

A basic categorial grammar is a tuple $(\Sigma, \text{Prim}, S, \triangleleft)$ where Σ is a finite set of symbols, Prim is a finite set of primitive types, and $S \in \text{Tp}(\text{Prim})$. The relation $\triangleleft$ is the lexicon, which relates types to $\text{symbols}(\triangleleft) \subseteq \text{Tp}(\text{Prim}) \times \Sigma$. Lexicons can be specified by listing a set of pairs like TYPE$\triangleleft$ symbol.

Categorial Grammar

Example: A grammar for English

- Basic Types:N, NP, S
- The lexicon for the sentence "the bad boy made that mess"

$$
\begin{aligned}
NP/N &\triangleleft \text{the} \\
NP/N &\triangleleft \text{that} \\
N &\triangleleft \text{boy} \\
N &\triangleleft \text{mess} \\
N/N &\triangleleft \text{bad} \\
(NP\backslash S)/NP &\triangleleft \text{made}
\end{aligned}
$$

An Example Parse

the	bad	boy	made	that	mess
NP/N,	N/N,	N,	(NP\ S)/NP,	NP/N,	N

$$NP/N,\ N/N,\ N,\ (S\backslash NP)/NP,\ \underbrace{NP/N,\ N}_{NP}$$

$$NP/N,\ N/N,\ N,\ \underbrace{(S\backslash NP)/NP,\quad NP}_{(S\backslash NP)}$$

$$NP/N,\ \underbrace{N/N,\ N}_{N},\quad (S\backslash NP)$$

$$\underbrace{NP/N,\quad N}_{NP},\quad (S\backslash NP)$$

$$\underbrace{\qquad\qquad\qquad}_{S}$$

.

The parse is **((the (bad boy)) (made (that mess)))**

Categorial Grammar

- Motivation for using CCGs
  - Categorial grammars of this form (having only function application rules) are equivalent in generative capacity to context-free grammars and are thus often considered inadequate for theories of natural language syntax

From CG to CCG

CCG is an **extension** of CG.
CCG has more rules:

- forward and backward type raising
- forward and backward composition

Coordination

Coordination: $(< \& >)$

$$X \quad conj \quad X \quad \Rightarrow \quad X$$

$$\frac{\dfrac{\dfrac{\dfrac{\text{I}}{NP} \quad \dfrac{\text{loathe}}{(S\backslash NP)/NP} \quad \dfrac{\text{and}}{CONJ} \quad \dfrac{\text{detest}}{(S\backslash NP)/NP}}{(S\backslash NP)/NP}\,{<}\&{>} \quad \dfrac{\text{opera}}{NP}}{S\backslash NP}\,{>}}{S}\,{<}$$

Type Raising

- CCG includes type-raising rules, which turn arguments into functions over functions over such arguments
- Forward type raising

$$\frac{X}{Y/(Y\backslash X)}\,\mathrm{T}$$

- Example

$$\frac{\text{John}}{\dfrac{\text{NP}}{\text{S}/(\text{S}\backslash\ \text{NP})}\,\mathrm{T}}$$

Multiple Derivations

Q1: *I know what restaurant serves French food, but what restaurant serves Italian food?*

A1:

Babbo	serves	Italian food.
NP	S\NP/NP	NP
	S\NP (>)	

Q2: *I know what kind of food Pierre's serves, but what kind of food does Babbo serve?*

A2:

Babbo	serves	Italian food.
NP	S\NP/NP	NP
S/(S\NP) (>T)		
S/NP (>)		

Forward Composition

$$\frac{X/Y \qquad Y/Z}{X/Z}\,B$$

$$\frac{\dfrac{John}{S/(S\backslash NP)} \qquad \dfrac{likes}{(S\backslash NP)/NP}}{S/NP}\,B$$

CCG is more flexible

CCG generates more sentences:

- Object relative clauses
  "A restaurant that $[\textit{John likes}]_{S/NP}$"
- Right node raising
  $[\textit{John likes}]_{S/NP}$ but $[\textit{Charles hates}]_{S/NP}$ Giovanni's
- CCG allows one sentence to be derived in many ways reflecting different intonation patterns.