CS460/626 : Natural Language Processing/Speech, NLP and the Web (Lecture 17, 18, 19– Grammar; Constituency, Dependency)

Pushpak Bhattacharyya
CSE Dept.,
IIT Bombay
9th, 13th and 14th Feb, 2012
Grammar

- A finite set of rules
  - that generates *only and all* sentences of a language.
  - that assigns an appropriate structural description to each one.
Grammatical Analysis Techniques

- Two main devices

Breaking up a String
- Sequential
- Hierarchical
- Transformational

Labeling the Constituents
- Morphological
- Categorial
- Functional
Breaking up and Labeling

- Sequential Breaking up
  - Sequential Breaking up and Morphological Labeling
  - Sequential Breaking up and Categorial Labeling
  - Sequential Breaking up and Functional Labeling

- Hierarchical Breaking up
  - Hierarchical Breaking up and Categorial Labeling
  - Hierarchical Breaking up and Functional Labeling
Sequential Breaking up

- That student solved the problems.

that + student + solve + ed + the + problem + s
Sequential Breaking up and Morphological Labeling

- That student solved the problems.

Diagram:

```
that       student       solve       ed       the       problem       s
  word      word         stem        affix     word       stem        affix
```
Sequential Breaking up and Categorial Labeling

- This boy can solve the problem.

- They called her a taxi.
Sequential Breaking up and Functional Labeling

They called her a taxi

<table>
<thead>
<tr>
<th>Subject</th>
<th>Verbal</th>
<th>Direct Object</th>
<th>Indirect Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>They</td>
<td>called</td>
<td>her</td>
<td>a taxi</td>
</tr>
</tbody>
</table>
Hierarchical Breaking up

- Old men and women

```
Old men and women

Old
men and women

men and women

men and women

Old men
and
women

Old
men
```
Hierarchical Breaking up and Categorial Labeling

- Poor John ran away.
Hierarchical Breaking up and Functional Labeling

• Immediate Constituent (IC) Analysis
• Construction types in terms of the function of the constituents:
  – Predication  (subject + predicate)
  – Modification  (modifier + head)
  – Complementation (verbal + complement)
  – Subordination (subordinator + dependent unit)
  – Coordination (independent unit + coordinator)
Predication

- \([\text{Birds}]_{\text{subject}} \ [\text{fly}]_{\text{predicate}}\)
Modification

- \([A]_{\text{modifier}} [\text{flower}]_{\text{head}}\)
- \(\text{John} \ [\text{slept}]_{\text{head}} [\text{in the room}]_{\text{modifier}}\)
Complementation

- He [saw] verbal [a lake] complement

Diagram:
- S
  - Subject
    - He
  - Predicate
    - Verbal
      - saw
    - Complement
      - a lake
Subordination

- \textit{John slept [in]_{subordinator} [the room]_{dependent unit}}
Coordination

- [John came in time] independent unit [but] coordinator independent unit
- [Mary was not ready]

S

Independent Unit | Coordinator | Independent Unit

John came in time | but | Mary was not ready
An Example

- In the morning, the sky looked much brighter.
Hierarchical Breaking up and Categorial / Functional Labeling

- Hierarchical Breaking up coupled with Categorial / Functional Labeling is a very powerful device.
- But there are ambiguities which demand something more powerful.
- *E.g.*, *Love of God*
  - *Someone loves God*
  - *God loves someone*
Hierarchical Breaking up

Categorial Labeling

Love of God

Noun Phrase

love

Prepositional Phrase

of

God

Functional Labeling

Love of God

Head

love

Modifier

of

God
Types of Generative Grammar

- Finite State Model
  (sequential)

- Phrase Structure Model
  (sequential + hierarchical) + (categorial)

- Transformational Model
  (sequential + hierarchical + transformational) +
  (categorial + functional)
Phrase Structure Grammar (PSG)

A phrase-structure grammar $G$ consists of a four tuple $(V, T, S, P)$, where

- $V$ is a finite set of *alphabets* (or *vocabulary*)
- $T$ is a finite set of terminal symbols: $T \subset V$
  - *E.g.*, *student*, *sing*, *etc.*
- $S$ is a distinguished non-terminal symbol, also called *start symbol*: $S \in V$
- $P$ is a set of productions.
Noun Phrases

- John
- the student
- the intelligent student

```
NP
  N
  John

NP
  Det
  the

NP
  N
  student

NP
  Det
  the
  AdjP
  intelligent

NP
  N
  student
```
Noun Phrase

- his first five PhD students

Diagram:
```
   NP
  / \  /   /
 Det Ord Quant N  N
  his first five PhD students
```
Noun Phrase

- The five best students of my class
Verb Phrases

- can sing
- can hit the ball
Verb Phrase

- Can give a flower to Mary
Verb Phrase

- may make John the chairman
Verb Phrase

- may find the book very interesting
Prepositional Phrases

- in the classroom
- near the river

Diagram:

```
  PP
   /\  
  P   NP
         /
        in
        the classroom

  PP
   /\  
  P   NP
         /
        near
        the river
```
Adjective Phrases

- intelligent
- very honest
- fond of sweets

![Diagram of adjective phrases]

- AP: Adjective Phrase
- A: Adjective
- Degree: Degree of Comparison
- PP: Prepositional Phrase
- of sweets
very worried that she might have done badly in the assignment
X Bar Theory

Consider the sentence:

*His first five PhD Students*

Is the structure for this deeper as shown?

Test: **One Replacement**

→ *His first five PhD Students and not the second ones*

→ *His first five PhD Students and not the M.Tech. ones*

→ *His first five PhD Students and not the six ones*
How do we know if a phrase XP (X= N/V/A) is of a particular type?

- **Tests**
  - Can XP be replaced with an elementary X (N/V/A), i.e. can XP be replaced with a single *unambiguous* word that fits perfectly in its place?
    - Eg: *He is very worried that she might fail*
    - *He is XP*
    - *He is honest/worried/running*: Adjectives

- *Head* can decide the type of the phrase
  - Determining the Head is a complex task

- Consider other languages: Hindi: उनका रोज़ यहाँ आना मुझे पसंद नहीं है
  - उनका रोज़ यहाँ आना can be replaced with आम and hence the whole thing is an NP
Consider the Grammar \((L=(ab)^+)\):

- \(S \rightarrow ab\) \(P=1/3\)
- \(S \rightarrow SS\) \(P=2/3\)

For \(l=2\):
- \(ab\)
- \(P_{l=2} = 1/3\)

For \(l=4\):
- \(abab\)
- \(P_{l=4} = 2/3 \times (1/3)^2\)

For \(l=6\):
- \(ababab\)
- \(P_{l=6} = 2 \times (2/3)^2 \times (1/3)^3\)
This can be solved to obtain a recursive function:

- \( P_{l=2} = \frac{1}{3} \)
- \( P_{l=4} = \frac{2}{3} \times \frac{1}{3} \times P_{l=2} \)
- \( P_{l=6} = 2 \times \frac{2}{3} \times \frac{1}{3} \times P_{l=4} \)
- \( P_{l=8} = \left(2 \times \frac{2}{3} \times \frac{1}{3} \times P_{l=6}\right) + \left(\frac{2}{3} \times P_{l=4}^2\right) \)

- \( P_{l=2n} = f(P_{l=2n-2}) \)

- \( \sum_{n=1}^{\infty} P_{l=2n} = \frac{1}{2} \)

Shouldn’t this be 1? Why? Why not?
Phrase Structure Rules

- The boy hit the ball.

- Rewrite Rules:
  (i) $S \rightarrow NP \ VP$
  (ii) $NP \rightarrow Det \ N$
  (iii) $VP \rightarrow V \ NP$
  (iv) $Det \rightarrow \text{the}$
  (v) $N \rightarrow \text{man, ball}$
  (v) $V \rightarrow \text{hit}$

- We interpret each rule $X \rightarrow Y$ as the instruction *rewrite X as Y*. 
Derivation

- The boy hit the ball.

<table>
<thead>
<tr>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP + VP (i)</td>
</tr>
<tr>
<td>Det + N + VP (ii)</td>
</tr>
<tr>
<td>Det + N + V + NP (iii)</td>
</tr>
<tr>
<td>The + N + V + NP (iv)</td>
</tr>
<tr>
<td>The + boy + V + NP (v)</td>
</tr>
<tr>
<td>The + boy + hit + NP (vi)</td>
</tr>
<tr>
<td>The + boy + hit + Det + N (ii)</td>
</tr>
<tr>
<td>The + boy + hit + the + N (iv)</td>
</tr>
<tr>
<td>The + boy + hit + the + ball (v)</td>
</tr>
</tbody>
</table>
The boy hit the ball.
John wrote those words in the Book of Proverbs.
• John wrote those words in the Book of Proverbs.

[John/NNP ]
wrote/VBD
[ those/DT words/NNS ]
in/IN
[ the/DT Book/NN ]
of/IN
[ Proverbs/NNS ]
John wrote those words in the Book of Proverbs.

(S (NP-SBJ (NP John))
  (VP wrote
    (NP those words)
    (PP-LOC in
      (NP (NP-TTL (NP the Book)
         (PP of
           (NP Proverbs))))))
Official trading in the shares will start in Paris on Nov 6.
• Official trading in the shares will start in Paris on Nov 6.

[ Official/JJ trading/NN ]
in/IN
[ the/DT shares/NNS ]
will/MD start/VB in/IN
[ Paris/NNP ]
on/IN
[ Nov./NNP 6/CD ]
Official trading in the shares will start in Paris on Nov 6.
Penn POS Tag Sset

- Adjective: JJ
- Adverb: RB
- Cardinal Number: CD
- Determiner: DT
- Preposition: IN
- Coordinating Conjunction: CC
- Subordinating Conjunction: IN
- Singular Noun: NN
- Plural Noun: NNS
- Personal Pronoun: PP
- Proper Noun: NP
- Verb base form: VB
- Modal verb: MD
- Verb (3sg Pres): VBZ
- Wh-determiner: WDT
- Wh-pronoun: WP
Difference between constituency and dependency
Constituency Grammar

- Categorical
- Uses part of speech
- Context Free Grammar (CFG)
- Basic elements → Phrases
Dependency Grammar

- Functional
- Context Free Grammar
- Basic elements → Units of Predication/Modification/Complementation/Subordination/Co-ordination
Bridge between Constituency and Dependency parse

- Constituency uses phrases
- Dependencies consist of Head-modifier combination
- *This is a cricket bat.*
  - *Cricket* (Category: N, Functional: Adj)
  - *Bat* (Category: N, Functional: N)
- For languages which are free word order we use dependency parser to uncover the relations between the words.
- *Raam ne Shaam ko dekha.* (Ram saw Shyam)
- *Shaam ko Ram ne dekha.* (Ram saw Shyam)
- Case markers cling to the nouns they subordinate
Example of CG and DG output

*Birds Fly.*

```
S -> NP -> N -> Birds
V -> fly
```

```
S -> Subject -> Birds
V -> fly
```
Some probabilistic parsers and why they are used

- Stanford, Collins, Charniack, RASP
- Why Probabilistic parsers
  - For a single sentence we can have multiple parses.
  - Probability for the parse is calculated and then the parse with the highest probability is selected.
  - This is needed in many applications of NLP, that need parsing.
A note on Language Modeling

- Example sentence
  - "^ The tortoise beat the hare in the race."

<table>
<thead>
<tr>
<th>N-gram (n=3)</th>
<th>CFG</th>
<th>Probabilistic CFG</th>
<th>Dependency Grammar</th>
<th>Prob. DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>^ the tortoise 5*10^{-3}</td>
<td>S-&gt; NP VP</td>
<td>S-&gt;NP VP 1.0</td>
<td>Semantic Roles</td>
<td>Semantic Roles with probabilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>agt, obj, sen, etc.</td>
<td></td>
</tr>
<tr>
<td>the tortoise beat</td>
<td>NP-&gt;DT N</td>
<td>NP-&gt;DT N 0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3*10^{-2}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tortoise beat the</td>
<td>VP-&gt;V NP PP</td>
<td>VP-&gt;V NP PP 0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7*10^{-5}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beat the hare</td>
<td>PP-&gt; P NP</td>
<td>PP-&gt; P NP 1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5*10^{-6}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Tortoise beat the hare in the race
UNL Expression

![Diagram of UNL Expression]

The diagram represents the UNL Expression with the following components:

- **beat@past**
  - **agt**: tortoise@def
  - **obj**: hare@def
  - **scn (scene)**: race@def
Purpose of LM

- Prediction of next word (Speech Processing)
- Language Identification (for same script)
- Belongingness check (parsing)
- \( P(NP->DT \; N) \) means what is the probability that the ‘YIELD’ of the non terminal NP is DT N