Phrases and Parsing

Sarmad Hussain

Important Note

Much of this material is from:

Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition

By Daniel Jurafsky, James H. Martin
Published by Pearson Prentice Hall, 2008
ISBN 0131873210, 9780131873216

These slides are for Computational Linguistics courses at NUCES; For course use only and not for further circulation or reuse due to possible copyright violations; Please purchase original book
Phrase

• Words are not “flatly” arranged in a sentence but are grouped in smaller parts called phrases

The old man on the street walked feebly ahead

Three Core Concepts

• Constituency
• Grammatical Roles/Relations
• Subcategorization
Constituency

- Constituency tells us how words are grouped within a sentence
  - Words group together to form a constituent or a phrase
  - Constituency is hierarchical
  - Constituents are named after their heads
    - Noun Phrase, NP, has noun as a head
      - Car, sports car, sleek sports car
      - علي، بوزها تخص

Grammatical Relations

- Constituents on a sentence play different roles, and are related to each other
- These roles are called grammatical relations

The boy gave his teacher the book

- NP Subject
- NP Secondary Object
- NP Object
Sub-categorization

- How are roles related within a sentence?
- A main Predicate (Verb) may need a Subject (NP) and an Object (NP) and an optional Adjunct (PP)

The boy **gave** the book to the teacher

Context Free Grammar

- Used for defining constituency
- CFG:

  \[
  N \rightarrow \text{set of non-terminal symbols} \\
  \Sigma \rightarrow \text{set of terminal symbols} \\
  P \rightarrow \text{set of productions } N \rightarrow \alpha \\
  \text{where } \alpha \text{ is a string from } (\Sigma U N)^* \\
  S \rightarrow \text{start symbol, belongs to } N
  \]
English Grammar

• Sentence
  – Declarative
    • S → NP VP
      • The boy ate the breakfast
  – Imperative
    • S → VP
      • Eat the breakfast
  – Questions
    • Yes-no: S → Aux NP VP
      • Can the boy eat the breakfast
    • Wh- : S → Wh- NP VP
      • Which boy ate the breakfast

Noun Phrase

• Nominal
  – Nom → Noun
    – boys

• Pre-Nominal Modifiers
  – NP → (Det) (Card) (Ord) (Quant) (AP) Nom
    – The first few big boys

• Post-Nominal Modifiers
  – NP → Nom GerundiveVP
    – The first few big boys arriving in the morning
  – NP → Nom RelClause
    – The first few big boys that arrived in the morning
Verb Phrase

- VP $\rightarrow$ Verb
  - I eat
- VP $\rightarrow$ Verb NP
  - I eat an apple
- VP $\rightarrow$ Verb NP PP
  - I eat an apple in the morning
- VP $\rightarrow$ Verb PP
  - I eat in the morning
Verb Sub-Categorization

- Verbs may require complements
  - Intransitive, Transitive, Ditransitive
- Sub-categorization is based on verb semantics
- A verb may have multiple sub-cat frames

<table>
<thead>
<tr>
<th>Frame</th>
<th>Verb</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\emptyset$</td>
<td>eat, sleep</td>
<td>I want to eat</td>
</tr>
<tr>
<td>$NP$</td>
<td>prefer, find, leave, show, give</td>
<td>Find $[NP$ the flight from Pittsburgh to Boston$]$; Show $[NP$ me$]$ $[NP$ airlines with flights from Pittsburgh$]$</td>
</tr>
<tr>
<td>$PP_{from} PP_{to}$</td>
<td>fly, travel</td>
<td>I would like to fly $[PP$ from Boston$]$ $[PP$ to Philadelphia$]$</td>
</tr>
<tr>
<td>$NP PP_{with}$</td>
<td>help, load,</td>
<td>Can you help $[NP$ me$]$ $[PP$ with a flight$]$</td>
</tr>
<tr>
<td>$VP_{to}$</td>
<td>prefer, want, need</td>
<td>I would prefer $[VP_{to}$ to go by United airlines$]$</td>
</tr>
<tr>
<td>$VP_{brst}$</td>
<td>can, would, might</td>
<td>I can $[VP_{brst}$ go from Boston$]$</td>
</tr>
<tr>
<td>$S$</td>
<td>mean</td>
<td>Does this mean $[S$ AA has a hub in Boston$]$?</td>
</tr>
</tbody>
</table>

Auxiliaries

- Categories
  - Modal
    - can, could, may, might, will, would, shall, ...
  - Perfect
    - have
  - Progressive
    - be
  - Passive
    - be
- Order
  - modal < perfect < progressive < passive
    - could have been sleeping
Other Phrases

• Adjectival Phrase
  – AP $\rightarrow$ ((Adj)$^+$ CC) Adj
  – big, fat and ugly monster

• Prepositional Phrase
  – PP $\rightarrow$ Prep. NP
  – in the morning, to the school

• Conjunctions
  – NP $\rightarrow$ NP CC NP
  – Generically, X $\rightarrow$ X CC X

Agreement

• Morphology of words depend on the context of other words

• Language choose different features for agreement
  – English agrees on person and number
  – Urdu agrees on person, number, gender, case and respect
  – Pashto agrees on person, number, gender and case
    – لﮑﮭﺎ + verb + singular + non-past + 2nd person + plural + respect level 3

• Agreement varies with word class
  – In Urdu nouns agree with case, number and gender
  – In Pashto,
    • number, person agreement in Verbs for 1st and 2nd person
    • number, person, gender agreement in Verbs for 3rd person
CFG and Agreement

• Sentence1S → pr1S VP1S
  – I go to the school
• Sentence1P → pr1P VP1P
  – We go to the school
• Sentence3S → pr3S VP3S
  – He goes to the school
• Sentence3P → pr3P VP3P
  – They go to the school

Treebanks

• Corpora marked with syntactic structures
• English: PENN Treebank, Arabic, Chinese...
Phrasal Heads

- Useful info for further processing

![Phrasal Heads Diagram](image)

- Rule based systems to identify phrasal heads
  - If the last word is tagged POS, return last word.
  - Else search from right to left for the first child which is an NN, NNP, NNPS, NX, POS, or JJ.
  - Else search from left to right for the first child which is an NP.
  - Else search from right to left for the first child which is a S, ADJP, or PRN.

Dependency Grammars

- Grammars focusing on functional relations rather than syntactic constituency
  - Incorporates semantic/deeper information
  - Better handles free-word-order languages

![Dependency Grammar Diagram](image)
Dependency Grammars

<table>
<thead>
<tr>
<th>Dependency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subj</td>
<td>syntactic subject</td>
</tr>
<tr>
<td>obj</td>
<td>direct object (incl. sentential complements)</td>
</tr>
<tr>
<td>dat</td>
<td>indirect object</td>
</tr>
<tr>
<td>pcomp</td>
<td>complement of a preposition</td>
</tr>
<tr>
<td>comp</td>
<td>predicate nominals (complements of copulas)</td>
</tr>
<tr>
<td>tmp</td>
<td>temporal adverbials</td>
</tr>
<tr>
<td>loc</td>
<td>location adverbials</td>
</tr>
<tr>
<td>attr</td>
<td>premodifying (attributive) nominals (genitives, etc.)</td>
</tr>
<tr>
<td>mod</td>
<td>nominal postmodifiers (prepositional phrases, etc.)</td>
</tr>
</tbody>
</table>

Urdu Grammar

• Sentences
  – Declarative
    • ائتمام نے انہوں کتاب دی
  – Interrogative
    • کیا ائتمام نے انہوں کتاب دی؟
    • ائتمام نے انہوں کتاب دی؟
  – Imperative
    • کتاب دو
    • کتاب دئیں
(Case)

- Nominative/Ergative
- Accusative
- Dative
- Genitive
- Instrumental
- Vocative

Noun Phrase

- Pre-modifiers
  - NP → (GenP) (Ord) (Quant) (Card) (AdjP) Nom | VPInf
  - NP → Nom RelClause
- Post-modifiers
  - NP → Nom RelClause
  - RelClause → S_w/o Subj | Obj
Verb Phrase

- VP → Verb AspAux* (Taux)
  - اسم کا کام کریا
  - اسم کا کام کریا تما
  - اسم کا کام کریا را روا
  - پہنچا ازاں پچلے بار بہت دوچ دول گے

Complex Predicates

- Complex Verbs
  - N Verb
    - اسم کا کام کریا
    - اسم کا کام کریا ویا
  - AdjP Verb
    - اسم کا کام کریا

25

26
Parsing

• Given a grammar, a lexicon, and an input sentence, Parsing would mean:
  – Find a sequence of rules, which start from S production
  – The rules expand to non-terminals which generate exactly the same amount of terminals as the input sentence, in the sequence of the input
• Parsing is a search through the set of rules provided by the grammar
• Parsing also creates a tree structure called the parse tree

Grammar and Parse Tree
Methods of Parsing

- Start from S and keep expanding all rules possible till possibilities are exhausted
  - Correct parses are those which lead to terminals in the input
- Start from terminals and keep reducing all rules until all possibilities are exhausted
  - Correct parses are those which end up in the symbol S

Top Down Parsing
Top Down vs. Bottom Up Parsing

- Top-Down parsing always starts from S productions, thus does not waste time exploring parses which may not result in S.
- Bottom-up parsing always starts from the terminals, thus does not waste time exploring parses which do not end with terminals.
Ambiguity in Parsing

• Like POS tagging, parsing process may also be ambiguous
  – Multiple structures are created causing structural ambiguity

Attachments Ambiguity
  – PP Attachment
    – I [shot an elephant] in my pajamas
    – I shot [an elephant in my pajamas]

Coordination Ambiguity
  – CC Attachment
    – Old [men and women]
    – [Old men] and [women]
Resolving the Ambiguity

• Ambiguity can be resolved by calculating the probability of a parse structure

• Prob. CFG will be discussed later

• First, let us look at parsing with ambiguity and storing all possibilities

Cocke-Kasami-Younger (CYK) Parsing Algorithm

• Based on table based Dynamic Programming
  – Solve partial problems to solve larger problem

• Higher level Algorithm
  – Convert grammar to CNF
  – Fill in table with partial parse information
  – Trace back from the end to find the complete parse
CYK Parsing: CNF

- Only two types of productions allowed
  - \( A \rightarrow B C \)
  - \( A \rightarrow w' \)

- Copy productions conforming to these forms
- Change other productions to this form
  - If: \( A \rightarrow B C D \), convert to:
    - \( A \rightarrow X_1 D \)
    - \( X_1 \rightarrow B C \)
  - If: \( A \rightarrow B \), combine with subsequent production
    - \( B \rightarrow C D E \)
    - \( A \rightarrow C D E \)

- Copy productions conforming to these forms
- Change other productions to this form
  - If: \( A \rightarrow B X_1 \), convert to:
    - \( X_1 \rightarrow C D \)

- Also possible

---

S → NP VP
S → Aux NP VP
S → VP

NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → Verb NP PP
VP → Verb PP
VP → VP PP
PP → Preposition NP

S → NP VP
S → X_1 VP
X_1 → Aux NP
S → book | include | prefer
S → Verb NP
S → X_2 VP
S → Verb PP
S → VP PP
NP → 1 | she | me
NP → TWA | Houston
NP → Det Nominal
Nominal → book | flight | meal
Nominal → Nominal Noun
Nominal → Nominal PP
VP → book | include
VP → Verb NP
VP → X_2 PP
X_2 → Verb NP
VP → Verb PP
VP → VP PP
PP → Preposition NP
### CYK Parsing: Table Creation

<table>
<thead>
<tr>
<th></th>
<th>Book</th>
<th>that</th>
<th>flight</th>
<th>through</th>
<th>Houston</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>S, VP, Nom, N, Verb</td>
<td>[0,1]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Det</td>
<td>[1,2]</td>
<td></td>
<td></td>
<td>S, VP</td>
</tr>
<tr>
<td>2</td>
<td>N, Nom</td>
<td>[2,3]</td>
<td>S, X2, VP</td>
<td>[0,3]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Prep</td>
<td>[3,4]</td>
<td>[0,4]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>PP</td>
<td>[3,5]</td>
<td>[1,4]</td>
<td></td>
<td>[1,5]</td>
</tr>
<tr>
<td>5</td>
<td>Nom</td>
<td>[2,4]</td>
<td></td>
<td></td>
<td>[2,5]</td>
</tr>
</tbody>
</table>

**Rules:**
- S → NP VP
- S → X1 VP
- X1 → Aux NP
- S → book | include | prefer
- S → Verb NP
- S → X2 PP
- S → Verb PP
- NP → I, she | me
- NP → TWA | Houston
- NP → Det Nominal
- Nominal → book | flight | meal
- Nominal → Nominal Noun
- Nominal → Nominal PP
- VP → book | include | prefer
- VP → Verb NP
- VP → X2 PP
- X2 → Verb NP
- VP → Verb PP
- VP → PP
- PP → Preposition NP
- Det → that | this | a
- Noun → book | flight | meat |
CYK Parsing: The Parse

- Store two pointers with each symbol in each cell which point back to the source

- Allow multiple instances of a Non-terminal in a cell if two productions are possible

CYK Parsing: Table Creation
Problem with CYK Parsing

• The parse tree gives a path which is not representative of the parse tree drawn by linguists
  – Not linguistically intuitive because of conversion to CNF
    • X1, X2, etc. have no linguistic motivation

• Can solve the issue by mapping the productions back to original grammar