Faculty of Computer Science, Dalhousie University *2-Dec-2025*

CSCI 4152/6509 — Natural Language Processing

Lecture 18: Syntax of Natural Languages

Location: Studley LSC-Psychology P5260 Instructor: Vlado Keselj
Time: 14:35 – 15:55

Previous Lecture

- P0 discussion (4): P-18
- Activation functions, softmax function
- Neural language model, RNN
- Stacked and bidirectional RNN
- LSTM, self-attention, transformers

  Part IV: Parsing (Syntactic Processing)
- Prolog introduction
  - unification and backtracking
  - variables, lists; examples: factorial, member

19 Natural Language Syntax

Slide notes:

Natural Language Syntax

- Syntax — NLP level of processing
  - Syntax = sentence structure; i.e., study of the phrase structure
- sýntaxis (Greek) — "setting out together, arrangement"
- Words are not randomly ordered — word order is important and non-trivial
- There are "free-order" languages (e.g., Latin, Russian), but they are not completely order free.
- Reading: Chapter 12 (JM book) or Ch.18 (JM on-line)

Phrase Structure and Dependency Structure

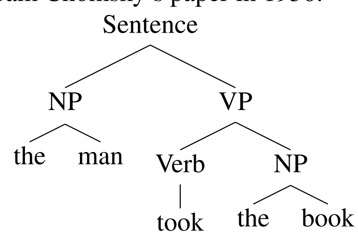
- Two ways of organizing sentence structure:
  - phrase structure
  - dependency structure
- Phrase structure
  - nested consecutive groupings of words
- Dependency structure
  - dependency relations between words
- The main NLP task at the syntax level: parsing
  - given a sentence, find the correct structure

Phrase Structure

- Phrase Structure Grammars or Context-Free Grammars
- A hierarchical view of sentence structure:
  - words form phrases
  - phrases form clauses
  - clauses form sentences
- Parsing: given a sentence find the context-free parse tree; a.k.a. phrase structure parse tree

Phrase Structure Parse Tree Examples

- Phrase Structure parse trees are also called Context-Free parse trees
- This example is from the seminal Noam Chomsky's paper in 1956:

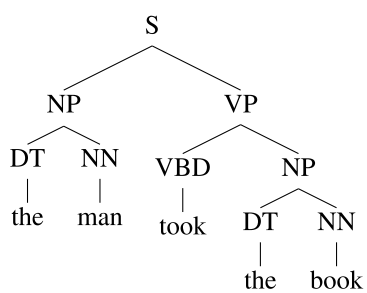


The above example is from the seminal article of Noam Chomsky, "Three models for the description of language" published in IRE Transactions on Information Theory in 1956.

If we follow more closely the Penn treebank tagset, we would rewrite the above parse tree as follows:

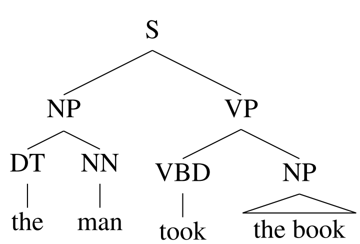
Parse Tree Examples (Penn treebank tagset)

- Using Penn treebank tagset:



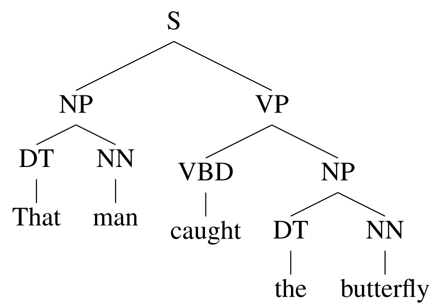
Parse Tree Examples ('triangle' notation)

- Sometimes we simplify a parse tree by ignoring a part of the structure, as in:



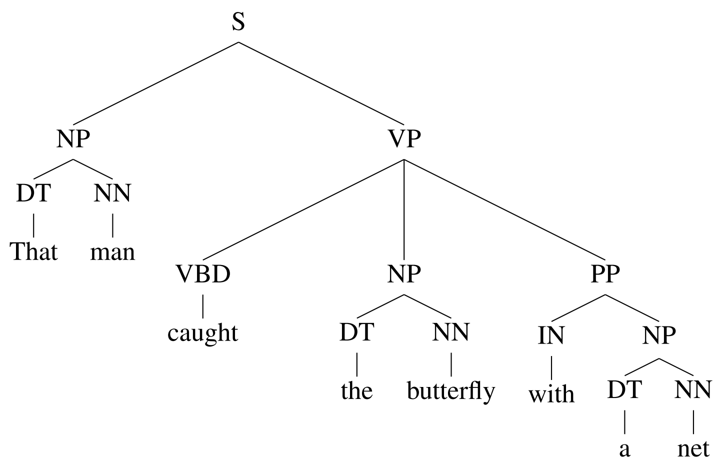
Parse Tree Example 2 (‘butterfly’)

- Another example:



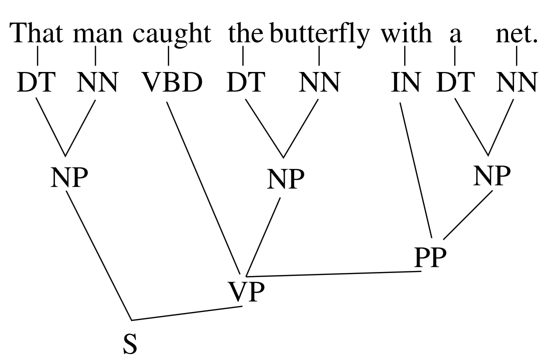
Parse Tree Example3 (‘butterfly’ extended)

- Extending the previous example:



Parse Tree Example (root bottom)

- Representing parse trees in the bottom-up direction:



Some Basic Notions in Context-Free Trees

- Context-free trees, also called phrase structure trees, parse trees, syntactic trees
- Node relations: root, leaf, parent (mother), child (daughter), sibling, ancestor, descendant, dominate
- Context-free grammar
- Consider for example the context-free grammar induced by the last parse tree shown

20 Context-Free Grammars (CFG) Review

Context-Free Grammars (CFG) Review

CFG is a tuple (V, T, P, S), where

- V is a finite set of **variables** or **non-terminals**; e.g., $V = \{S, NP, DT, NN, VP, VBD, PP, IN\}$
- T is a finite set of **terminals**, words, or lexemes; e.g., $T = \{\text{That, man, caught, the, butterfly, with, a, net}\}$
- P is a set of **rules** or **productions** in the form $X \to \alpha$, where $X \in V$ and $\alpha \in (V \cup T)^*$; e.g.,
  $P = \{S \to NP\ VP,\ \ NP \to DT\ NN,\ \ DT \to \text{That},\ \ NP \to \epsilon\}$
- S is the **start symbol** $S \in V$

Some Notions about CFGs

- CFG, also known as Phrase-Structure Grammar (PSG)
- Equivalent to BNF (Backus-Naur form)
- Idea from Wundt (1900), formally defined by Chomsky (1956) and Backus (1959)
- Typical notation (V, T, P, S); also (N, Σ, R, S)

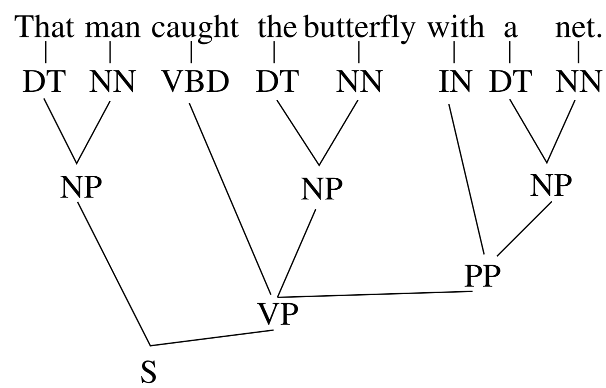
CFG (Context-Free Grammar) is also knows as the Phrase-Structure Grammar (PCG). It is usually referred to as CFG in the Formal Language theory and Computer Science in general, while the term PCG is used in some Computational Linguistic and Linguistic circles. The idea of CFG is traced to Wundt in 1900, but the exact created of the formalism is attributed to Noam Chomsky (1956), who worked in the area of Natural Language Processing, and also to John Backus, who worked in the area of Programming Languages.

CFG Derivations

CFG Derivations

- Direct derivation, derivation
- Example of a direct derivation: $S \Rightarrow NP\ VP$
- Example of a derivation (beginning of):
  $S \Rightarrow NP\ VP \Rightarrow DT\ NN\ VP \Rightarrow \text{That}\ NN\ VP \Rightarrow \ldots$
- Left-most and right-most derivation

Parse Tree Example (revisited)



Leftmost Derivation Example

$$
\begin{aligned}
S &\Rightarrow NP\ VP \Rightarrow DT\ NN\ VP \Rightarrow \text{That}\ NN\ VP \Rightarrow \text{That man}\ VP \\
&\Rightarrow \text{That man}\ VBD\ NP\ PP \\
&\Rightarrow \text{That man caught}\ NP\ PP \\
&\Rightarrow \text{That man caught}\ DT\ NN\ PP \\
&\Rightarrow \text{That man caught the}\ NN\ PP \\
&\Rightarrow \text{That man caught the butterfly}\ PP \\
&\Rightarrow \text{That man caught the butterfly}\ IN\ NP \\
&\Rightarrow \text{That man caught the butterfly with}\ NP \\
&\Rightarrow \text{That man caught the butterfly with}\ DT\ NN \\
&\Rightarrow \text{That man caught the butterfly with a}\ NN \\
&\Rightarrow \text{That man caught the butterfly with a net}
\end{aligned}
$$

Some Notions about CFGs (continued)

- Language generated by a CFG
- Context-Free languages
- Parsing task
- Ambiguous sentences
- Ambiguous grammars
- Inherently ambiguous languages

A language over a set of words (or terminals) in this context (i.e., the formal language context) is any set of strings of words, where a string of words is any finite, possibly empty, sequence of words.

The language generated by a CFG is the set of all strings of words that can be derived from the start symbol using a derivation.

A language is a context-free language, if there is a CFG that generates this language.

The parsing problem for a CFG is the problem of finding all parse trees of an arbitrary string of words, which may include an empty set of trees if the string does not belong to the language generated by the grammar.

Bracket Representation of a Parse Tree

```
(S (NP (DT That)
       (NN man))
   (VP (VBD caught)
       (NP (DT the)
           (NN butterfly))
       (PP (IN with)
           (NP (DT a)
               (NN net)
)  )   )   )
```

Some Notes on CFGs

– Left-hand side (lhs) and right-had side (rhs) of a production

$$\underbrace{S}_{lhs} \rightarrow \underbrace{NP\ VP}_{rhs}$$

– Empty rule (epsilon rule, epsilon production): $V \rightarrow \epsilon$
– Unit production: $A \rightarrow B$, where A and B are non-terminals
– Notational variations:
  – use of '|': $P \rightarrow N \mid A\,P$, instead of $P \rightarrow N,\ \ P \rightarrow A\ P$
  – BNF notation: $P ::= N \mid A\,P$
  – use of word 'opt': $NP\ \ ::=\ \ DT\ NN\ PP_{opt}$
  – or Kleene star: $NP\ \ ::=\ \ DT\ NN\ PP^{*}$

21 Context-Free Grammars for Natural Language

21.1 Typical Phrase Structure Rules in English

Typical Phrase Structure Rules in English

– We will cover some typical phrase structure rules
– Specific to English but also generalizable to other languages
– **Not** *all rules are covered,* but the general principles should be adopted

Typical Sentence Rules (S)

```
S -> NP VP           Declarative sentences, e.g.:
              I want a flight from Halifax to Chicago.
              The flight should be eleven a.m. tomorrow.

S -> VP              Imperative sentences, e.g.:
                     Show the lowest fare.
                     Delete the file.

S -> Aux NP VP       Yes-no questions, e.g.:
                   Do any of these flights have stops?
                   Can you give me some information for United?
                                            (pragmatic meaning)
```

```
S -> Wh-NP VP      Wh-subject questions, e.g.:
                   What airlines fly from Halifax?
                   Whose flights serve breakfast.

S -> Wh-NP Aux NP VP   Wh-non-subject questions, e.g.:
                  What flights do you have on Tuesday?
```

Noun Phrase (NP)

- typically: pronouns, proper nouns, or determiner-nominal construction
- some typical rules

```
NP -> PRP                     e.g.: you
NP -> NNP | NNPS              e.g.: Halifax
NP -> PDT? DT JJ* NN PP*
NP -> NN NN                   e.g.: computer science
```

- in the last rule, we use regular expression notation to describe a set of different rules
- example: all the various flights from Halifax to Toronto
- example 2: all the thick red books on the shelves in the library
- determiners and nominals
- modifiers before head noun and after head noun
- postmodifier phrases

```
NP -> DT JJ* NN RelC
```

Examples of determiners: *a* stop, *the* flights, *this* flight

The determiners can be more complex; e.g., they can consist of a noun phrase and posessive ending *'s*, such as "United's flight" and "Denver's mayor's mother's canceled flight".

However, we will not label those structures as determiners in a context-free parse trees.

Relative Clauses

- RelC — relative clause
- clause (sentence-like phrase) following a noun phrase
- example: gerundive relative clause: flights arriving after 5pm
- example: infinitive relative clause: flights to arrive tomorrow
- example: restrictive relative clause: flight that was canceled yesterday

Verb Phrase (VP)

- organizes arguments around the verb
- typical rules

```
VP -> Verb            intransitive verbs;
                      e.g.: disappear
VP -> Verb NP         transitive verbs:
                      e.g.: prefer a morning flight
VP -> Verb NP NP      ditransitive verbs:
                      e.g.: send me an email
```

```
VP -> Verb PP*          sentential complements
VP -> Verb NP PP*
VP -> Verb NP NP PP*
```

– sentential complements, e.g.: You said these were two flights that were the cheapest.

Prepositional Phrase (PP)

– Preposition (IN) relates a noun phrase to other word or phrase
– Prepositional Phrase (PP) consists of a preposition and the noun phrase which is an object of that preposition
– There is typically only one rule for the prepositional phrase:

```
PP -> IN NP
```

– examples: from Halifax, before tomorrow, in the city
– PP-attachment ambiguity

Adjective Phrase (ADJP)

– less common
– examples:
  – She is <u>very sure of herself.</u>
  – ... the <u>least expensive</u> fare ...

Adverbial Phrase (ADVP)

– Example:

```
(S (NP preliminary findings)
   (VP were reported
       (ADVP (NP a year) ago)))
```

– another example: years ago

Overview of Syntactic Tags

Tag	Description
S	Sentence
NP	Noun Phrase
VP	Verb Phrase
PP	Prepositional Phrase
ADJP	Adjective Phrase
ADVP	Adverbial Phrase

About Typical Rules

– Only some typical rules are presented
– For example: We see the cat, and you see a dog.
– The sentence could be described with: `S -> S CC S`
– Relative clauses are labeled in Penn treebank using SBAR ($\bar{S}$) non-terminal; e.g.:

```
(S (NP (NP Lorillard Inc.)
       ,
       (NP (NP the unit)
           (PP of (NP (ADJP New York-based)
                      Loews Corp.)))
       (SBAR that
             (S (NP *gap*)
                (VP makes (NP Kent cigarettes))))
       ,)
    (VP stopped (VP using (NP crocidolite))))
```

Heads and Dependency (heads-up)

- a phrase typically has a central word called head, while other words are direct or indirect dependents
- a head is also called a governor, although sometimes these concepts are considered somewhat different
- phases are usually called by their head; e.g., the head of a noun phrase is a noun