Syntax, Grammar, Constituents and Dependencies

Yoav Goldberg

(with slides by Michael Collins, Julia Hockenmaier)
What is grammar?
Sentences in natural language have structure.
Linguists create Linguistic Theories for defining this structure.
The parsing problem is recovering that structure.
What is grammar?

Grammar formalisms
(= linguists’ programming languages)
   A precise way to define and describe
   the structure of sentences.
   (N.B.: There are many different formalisms out there, which each define their
   own data structures and operations)

Specific grammars
(= linguists’ programs)
   Implementations (in a particular formalism) for a particular
   language (English, Chinese,....)
Can we define a program that generates all English sentences?

The number of sentences is infinite. But we need our program to be finite.
Overgeneration

English

Undergeneration

John saw Mary.
I ate sushi with tuna.
I want you to go there.
Did you go there?
I ate the cake that John had made for me yesterday
John made some cake.

John Mary saw.
with tuna sushi ate I.
Did you went there?
.....

(by Julia Hockenmaier)
Basic sentence structure

I eat sushi.

Noun (Subject)  Verb (Head)  Noun (Object)
A finite-state-automaton (FSA)
A Hidden Markov Model (HMM)

- Noun (Subject)
  - I, you, ...
- Verb (Head)
  - eat, drink
- Noun (Object)
  - sushi, ...

(by Julia Hockenmaier)
Words take arguments

I eat sushi. ✓
I eat sushi you. ???
I sleep sushi ???
I give sushi ???
I drink sushi ?

Subcategorization:
Intransitive verbs (sleep) take only a subject.
Transitive verbs (eat) take also one (direct) object.
Ditransitive verbs (give) take also one (indirect) object.

Selectional preferences:
The object of eat should be edible.
A better FSA

Noun (Subject) → Transitive Verb (Head) → Noun (Object)

Intransitive Verb (Head)
Previously

- Structure is a sequence.
- Each item can be tagged.
- We can mark some spans.
Previously

- Structure is a sequence.
- Each item can be tagged.
- We can mark some spans.

Today

- Hierarchical Structure.
Hierarchical Structure?
Language is recursive

-the ball
-the big ball
-the big, red ball
-the big, red, heavy ball

Adjectives can modify nouns. The number of modifiers/adjuncts a word can have is (in theory) unlimited.
Another FSA
Recursion can be more complex

the ball
the ball in the garden
the ball in the garden behind the house
the ball in the garden behind the house next to the school
....
What is the structure of a sentence?

Sentence structure is hierarchical:

A sentence consists of **words** *(I, eat, sushi, with, tuna)*
.. which form phrases or **constituents**: “sushi with tuna”

Sentence structure defines dependencies between words or phrases:

```
I [eat [sushi [with tuna]]]
```
Strong vs. weak generative capacity

Formal language theory:
  – defines language as string sets
  – is only concerned with generating these strings
    *(weak generative capacity)*

Formal/Theoretical syntax (in linguistics):
  – defines language as sets of strings with (hidden) structure
  – is also concerned with generating the right structures
    *(strong generative capacity)*
Structure

Example 1: math

3*2+5*3
Structure

Example 1: math

ADD

MUL + MUL

3 * 2 + 5 * 3
Example 1: math

\[ \text{ADD} \quad \text{MUL} \quad \text{MUL} \]

\[
\begin{array}{c}
3 \quad * \\
2 \\
\end{array}
+ \quad
\begin{array}{c}
5 \quad * \\
3 \\
\end{array}
\]

\[3 \cdot 2 + 5 \cdot 3\]
Constituency (Phrase Structure)

Trees

• Phrase structure organizes words into nested constituents
Constituency (Phrase Structure) Trees

- Phrase structure organizes words into nested constituents
- Linguists can, and do, argue about details
Constituency (Phrase Structure) Trees

- Phrase structure organizes words into nested constituents
- Linguists can, and do, argue about details

```
NP
  N Fed
VP
  V raises
  NP
    N interest
    N rates
```

We will talk more about constituents soon.
Programming Languages?

They also have structure. How does it differ from human's language structure?
Structure
Example 2: Language Data

Fruit flies like a banana
Structure
Example 2: Language Data

Constituency Structure

Fruit flies like a banana

Dependency Structure

like

flies banana

Fruit a

bana
Dependency Representation

The lawyer questioned the witness.
Dependency Representation

I heard that the lawyer questioned the witness under oath yesterday.
Dependency Representation

I heard that the lawyer questioned the witness under oath yesterday.
Dependency representation is very common. We will return to it in the future.
Constituency Structure

- In this class we concentrate on Constituency Parsing: mapping from sentences to trees with labeled nodes and the sentence words at the leaves.
Why is Parsing Interesting?

- It’s a first step towards understanding a text.
- Many other language tasks use sentence structure as their input.
Some things that are done with parse trees

- Grammar Checking
- Word Clustering
- Information Extraction
- Question Answering
- Sentence Simplification
- Machine Translation
- ...and more
Some things that are done with parse trees

- Grammar Checking
- **Word Clustering**
- Information Extraction
- Question Answering
- Sentence Simplification
- Machine Translation
- ...and more

Words in similar grammatical relations share meanings.
Some things that are done with parse trees

- Grammar Checking
- Word Clustering
- Information Extraction
- Question Answering
- Sentence Simplification
- Machine Translation
- ... and more

Extract factual relations from text

Answer questions
Some things that are done with parse trees

- Grammar Checking
- Word Clustering
- Information Extraction
- Question Answering
- Sentence Simplification
- Machine Translation
- ... and more

The first new product, ATF Protype, is a line of digital postscript typefaces that will be sold in packages of up to six fonts.

ATF Protype is a line of digital postscript typefaces will be sold in packages of up to six fonts.
Some things that are done with parse trees

- Grammar Checking
- Word Clustering
- Information Extraction
- Question Answering
- Sentence Simplification
- Machine Translation
- ...and more
Some things that are done with parse trees

- Grammar Checking
- Word Clustering
- Information Extraction
- Question Answering
- Sentence Simplification
- Machine Translation
- ...and more
Constituency (Phrase Structure) Trees

- Phrase structure organizes words into nested constituents

What is a constituent?
How do we know they exist?
Context-free grammars (CFGs) capture recursion

Language has complex constituents
(“the garden behind the house”)

Syntactically, these constituents behave just like simple ones.
(“behind the house” can always be omitted)

CFGs define nonterminal categories to capture equivalent constituents.
The black dog sat

is "dog" a constituent?

is "The black dog" a constituent?

is "the black" a constituent?

is "dog sat" a constituent?

why?
Substitution Test

Every word is a constituent.

If we can substitute a sequence of words by a single word, it is likely to be a constituent.

In particular, pronouns.

(based on book chapter by Beatrice Santorini)
Substitution Test

(1) a. The little boy fed the cat. → He fed her.
   b. Black cats detest green peas. → They detest them.

(based on book chapter by Beatrice Santorini)
Substitution Test

(2) a. The little boy from next door fed the cat without a tail.
   b. These black cats detest those green peas.

(based on book chapter by Beatrice Santorini)
Substitution Test

(2) a. The little boy from next door fed the cat without a tail.
    b. These black cats detest those green peas.

   → * He from next door fed her without a tail.
   → * These they detest those them.

(based on book chapter by Beatrice Santorini)
Substitution Test

(3) a. The little boy from next door fed the cat without a tail.
   b. These black cats detest those green peas.

   $\rightarrow$ He fed her.
   $\rightarrow$ They detest them.

(based on book chapter by Beatrice Santorini)
Substitution Test

adverbs (here/there) can also be used.

(4) a. Put it on the table. → Put it there.
    b. Put it over on the table. → Put it over there.
    c. Put it over on the table. → Put it there.

(5) a. Put it on the table that's by the door. → * Put it there that's by the door.
    b. Put it over on the table that's by the door. → * Put it over there that's by the door.
    c. Put it over on the table that's by the door. → * Put it there that's by the door.
(6) a. I am very happy, and Linda is so, too.
   b. I am very fond of Lukas, and Linda is so, too.
   c. I am very fond of my nephew, * and Linda is so of her niece.

(based on book chapter by Beatrice Santorini)
Substitution Test

'so', 'it', instead of a subordinate clause.

(7) a. I { know, suspect } that they're invited. → I { know, suspect } it.
    b. I { imagine, think } that they're invited. → I { imagine, think } so.

(based on book chapter by Beatrice Santorini)
Substitution Test

Substitution tests are a good indicator of constituency.

But they may fail, also for real constituents.

There are other tests: movement, short answers, it clefts

We will briefly cover movement.
See more in the Santorini's chapter.

(based on book chapter by Beatrice Santorini)
Movement Test

Can we move the sequence to a different position in the sentence?

(based on book chapter by Beatrice Santorini)
Movement Test

(8) a. I fed the cats.

b. I fed the cats with long, fluffy tails.

→ The cats, I fed ___. (The dogs, I didn't.)

→ The cats with long, fluffy tails, I fed ___. (The other cats, I didn't.)

(based on book chapter by Beatrice Santorin)i
Movement Test

(9) a. Prepositional phrase: The cat strolled across the porch with a confident air.
b. Adjective phrase: Ali Baba returned from his travels wiser than before.
c. Adverb phrase: They arrived at the concert hall more quickly than they had expected.

→ With a confident air, the cat strolled across the porch ___.
→ Wiser than before, Ali Baba returned from his travels ___.
→ More quickly than they had expected, they arrived at the concert hall ___.

(based on book chapter by Beatrice Santorini)
Movement Test

(10) a. I fed the cats with long, fluffy tails.
   b. The cat strolled across the porch with a confident air.
   c. Ali Baba returned from his travels wiser than before.
   d. They arrived at the concert hall more quickly than they had expected.

(based on book chapter by Beatrice Santorini)
Movement Test

(10) a. I fed the cats with long, fluffy tails.
    b. The cat strolled across the porch with a confident air.
    c. Ali Baba returned from his travels wiser than before.
    d. They arrived at the concert hall more quickly than they had expected.

→ * The cats, I fed ___ with long, fluffy tails.¹
→ * With a, the cat strolled across the porch ___ confident air.
→ * Wiser than, Ali Baba returned from his travels ___ before.
→ * More quickly than they, they arrived at the concert hall ___ had expected.
See more tests in the notes.

I hope that you are convinced that constituents are "real".

You need to know how to identify them.
Parsing: recovering the constituents of a sentence.
Fat people eat candy
Why is parsing hard?

Ambiguity

Fat people eat candy

S

NP   VP

Adj   Nn   Vb   NP

Fat   people   eat   Nn
candy
Why is parsing hard?

Ambiguity

Fat people eat candy

Fat people eat accumulates

Diagram:

S
  NP
  Adj
  Fat
  Nn
  people
  Vb
  eat
  NP
  Nn
  candy
Why is parsing hard?

Ambiguity

Fat people eat candy

Fat people eat accumulates
Why is parsing hard?
Real Sentences are long... 

“Former Beatle Paul McCartney today was ordered to pay nearly $50M to his estranged wife as their bitter divorce battle came to an end.”

“Welcome to our Columbus hotels guide, where you’ll find honest, concise hotel reviews, all discounts, a lowest rate guarantee, and no booking fees.”
Context Free Grammars

A context free grammar $G = (N, \Sigma, R, S)$ where:

- $N$ is a set of non-terminal symbols
- $\Sigma$ is a set of terminal symbols
- $R$ is a set of rules of the form $X \rightarrow Y_1 Y_2 \cdots Y_n$ for $n \geq 0$, $X \in N$, $Y_i \in (N \cup \Sigma)$
- $S \in N$ is a special start symbol
Context Free Grammars

a simple grammar

\[ N = \{ S, NP, VP, Adj, Det, Vb, Noun \} \]
\[ \Sigma = \{ fruit, flies, like, a, banana, tomato, angry \} \]
\[ S = 'S' \]
\[ R = \]

\[ S \rightarrow NP \ VP \]
\[ NP \rightarrow Adj \ Noun \]
\[ NP \rightarrow Det \ Noun \]
\[ VP \rightarrow Vb \ NP \]
\[ Adj \rightarrow fruit \]
\[ Noun \rightarrow flies \]
\[ Vb \rightarrow like \]
\[ Det \rightarrow a \]
\[ Noun \rightarrow banana \]
\[ Noun \rightarrow tomato \]
\[ Adj \rightarrow angry \]
Left-most derivation is a sequence of strings $s_1, \cdots, s_n$ where

- $s_1 = S$ the start symbol
- $s_n \in \Sigma^*$, meaning $s_n$ is only terminal symbols
- Each $s_i$ for $i = 2 \cdots n$ is derived from $s_{i-1}$ by picking the left-most non-terminal $X$ in $s_{i-1}$ and replacing it by some $\beta$ where $X \rightarrow \beta$ is a rule in $R$.

For example: $[S],[NP \ VP],[Adj \ Noun \ VP], [fruit \ Noun \ VP], [fruit \ flies \ VP], [fruit \ flies \ Vb \ NP], [fruit \ flies \ like \ NP], [fruit \ flies \ like \ Det \ Noun], [fruit \ flies \ like \ a], [fruit \ flies \ like \ a \ banana]$
Left-most derivation example

S

The resulting derivation can be written as a tree. Many trees can be generated.
Left-most derivation example

S
NP VP

S → NP VP

The resulting derivation can be written as a tree.

Many trees can be generated.
Left-most derivation example

S
NP VP
Adj Noun VP

NP → Adj Noun
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP

Adj → fruit
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP

Noun → flies
Left-most derivation example

S
NP VP
  Adj Noun VP
  fruit Noun VP
  fruit flies VP
  fruit flies Vb NP

VP → Vb NP
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP
fruit flies Vb NP
fruit flies like NP

Vb → like
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP
fruit flies Vb NP
fruit flies like NP
fruit flies like Det Noun

NP → Det Noun
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP
fruit flies Vb NP
fruit flies like NP
fruit flies like Det Noun
fruit flies like a Noun

Det → a
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP
fruit flies Vb NP
fruit flies like NP
fruit flies like Det Noun
fruit flies like a Noun
fruit flies like a banana

Noun → banana
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP
fruit flies Vb NP
fruit flies like NP
fruit flies like Det Noun
fruit flies like a Noun
fruit flies like a banana

- The resulting derivation can be written as a tree.
Left-most derivation example

S
NP VP
Adj Noun VP
fruit Noun VP
fruit flies VP
fruit flies Vb NP
fruit flies like NP
fruit flies like Det Noun
fruit flies like a Noun
fruit flies like a banana

- The resulting derivation can be written as a tree.
- Many trees can be generated.
a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

- 

Adj → fruit
Noun → flies
Vb → like
Det → a
Noun → banana
Noun → tomato
Adj → angry

...
Context Free Grammars

a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

Adj → fruit
Noun → flies
Vb → like
Det → a
Noun → banana
Adj → angry

Example

S
   /\        /
  NP  VP  \
  /\    /\  /
Adj  Noun Vb NP
  \  /  \
  Fruit Flies like Det Noun
         /\   /\  /
         a  banana
a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

-  
  Adj → fruit
  Noun → flies
  Vb → like
  Det → a
  Noun → banana
  Noun → tomato
  Adj → angry

...
a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

- Adj → fruit
Noun → flies
Vb → like
Det → a
Noun → banana
Noun → tomato
Adj → angry

Example

S
  NP  VP
    Adj  Noun  Vb  NP
      Angry  Flies  like  Det  Noun
                   a  tomato
a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

- 
Adj → fruit
Noun → flies
Vb → like
Det → a
Noun → banana
Noun → tomato
Adj → angry

Example

S
----------
/      \\      \\
NP      VP
----------
/      \\      \\
Adj     Noun     Vb
----------
/      \\      \\
Angry banana like
          /      \\      \\
Det      Noun     a
----------
/      \\      \\
a tomato
Context Free Grammars

a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

- 
  Adj → fruit
  Noun → flies
  Vb → like
  Det → a
  Noun → banana
  Noun → tomato
  Adj → angry

...
a simple grammar

S → NP VP
NP → Adj Noun
NP → Det Noun
VP → Vb NP

-  Adj → fruit
Noun → flies
Vb → like
Det → a
Noun → banana
Noun → tomato
Adj → angry

...
A Brief Introduction to English Syntax
Product Details (from Amazon)
Hardcover: 1779 pages
Publisher: Longman; 2nd Revised edition
Language: English
ISBN-10: 0582517346
Product Dimensions: 8.4 x 2.4 x 10 inches
Shipping Weight: 4.6 pounds
A Brief Overview of English Syntax

Parts of Speech (tags from the Brown corpus):

- **Nouns**
  - NN = singular noun  e.g., man, dog, park
  - NNS = plural noun  e.g., telescopes, houses, buildings
  - NNP = proper noun  e.g., Smith, Gates, IBM
- **Determiners**
  - DT = determiner  e.g., the, a, some, every
- **Adjectives**
  - JJ = adjective  e.g., red, green, large, idealistic
A Fragment of a Noun Phrase Grammar

\[
\begin{align*}
N & \Rightarrow NN \\
\bar{N} & \Rightarrow NN \bar{N} \\
\bar{N} & \Rightarrow JJ \bar{N} \\
\bar{N} & \Rightarrow \bar{N} \bar{N} \\
NP & \Rightarrow DT \bar{N} \\
\end{align*}
\]

<table>
<thead>
<tr>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN \Rightarrow box</td>
<td>the box</td>
</tr>
<tr>
<td>NN \Rightarrow car</td>
<td>a car</td>
</tr>
<tr>
<td>NN \Rightarrow mechanic</td>
<td>the pigeon car</td>
</tr>
<tr>
<td>NN \Rightarrow pigeon</td>
<td>the fast metal pigeon</td>
</tr>
<tr>
<td>DT \Rightarrow the</td>
<td></td>
</tr>
<tr>
<td>DT \Rightarrow a</td>
<td></td>
</tr>
</tbody>
</table>

(by Mike Collins)
Prepositions, and Prepositional Phrases

- Prepositions
  \[ \text{IN} = \text{preposition} \quad \text{e.g., of, in, out, beside, as} \]
An Extended Grammar

<table>
<thead>
<tr>
<th>Ň</th>
<th>⇒</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ň</td>
<td>⇒</td>
<td>NN  Ň</td>
</tr>
<tr>
<td>Ň</td>
<td>⇒</td>
<td>JJ  Ň</td>
</tr>
<tr>
<td>Ň</td>
<td>⇒</td>
<td>Ň  Ň</td>
</tr>
<tr>
<td>NP</td>
<td>⇒</td>
<td>DT  Ň</td>
</tr>
<tr>
<td>PP</td>
<td>⇒</td>
<td>IN  NP</td>
</tr>
<tr>
<td>Ň</td>
<td>⇒</td>
<td>Ň  PP</td>
</tr>
</tbody>
</table>
An Extended Grammar

Generates:
in a box, under the box, the fast car mechanic under the pigeon in the box, ...
Verbs, Verb Phrases, and Sentences

- Basic Verb Types
  - $V_{i}$ = Intransitive verb e.g., sleeps, walks, laughs
  - $V_{t}$ = Transitive verb e.g., sees, saw, likes
  - $V_{d}$ = Ditransitive verb e.g., gave

- Basic VP Rules
  - $VP \rightarrow V_{i}$
  - $VP \rightarrow V_{t} \ NP$
  - $VP \rightarrow V_{d} \ NP \ NP$

- Basic S Rule
  - $S \rightarrow NP \ VP$

Examples of VP:
sleeps, walks, likes the mechanic, gave the mechanic the fast car

Examples of S:
the man sleeps, the dog walks, the dog gave the mechanic the fast car

(by Mike Collins)
A new rule: \( VP \rightarrow VP \text{ PP} \)

New examples of \( VP \):
sleeps in the car, walks like the mechanic, gave the mechanic the fast car on Tuesday, ...
Complementizers, and SBARs

- Complementizers
  \[ \text{COMP} \equiv \text{complementizer} \quad \text{e.g., that} \]

- SBAR
  \[ \text{SBAR} \rightarrow \text{COMP} \quad \text{S} \]

**Examples:**
that the man sleeps, that the mechanic saw the dog . . .
More Verbs

- New Verb Types
  - V[5]  e.g., said, reported
  - V[6]  e.g., told, informed
  - V[7]  e.g., bet

- New VP Rules
  - VP  →  V[5]   SBAR
  - VP  →  V[6]   NP   SBAR
  - VP  →  V[7]   NP   NP   SBAR

Examples of New VPs:
said that the man sleeps
told the dog that the mechanic likes the pigeon
bet the pigeon $50 that the mechanic owns a fast car

(by Mike Collins)
Coordination

- A New Part-of-Speech:
  \[ CC = \text{Coordinator} \quad \text{e.g., and, or, but} \]

- New Rules
  \[
  \begin{align*}
  \text{NP} & \rightarrow \text{NP} \quad CC \quad \text{NP} \\
  \bar{\text{N}} & \rightarrow \bar{\text{N}} \quad CC \quad \bar{\text{N}} \\
  \text{VP} & \rightarrow \text{VP} \quad CC \quad \text{VP} \\
  \text{S} & \rightarrow \text{S} \quad CC \quad \text{S} \\
  \text{SBAR} & \rightarrow \text{SBAR} \quad CC \quad \text{SBAR}
  \end{align*}
  \]
We’ve Only Scratched the Surface...

- Agreement
  
  The dogs laugh vs. The dog laughs

- Wh-movement
  
  The dog that the cat liked ___

- Active vs. passive
  
  The dog saw the cat vs.
  The cat was seen by the dog

- If you’re interested in reading more:
  
CFGs and center embedding

The mouse ate the corn.
The mouse that the snake ate ate the corn.
The mouse that the snake that the hawk ate ate ate the corn.

These sentences are all grammatical. They can be generated by a CFG:

\[
S \rightarrow NP \ VP \\
NP \rightarrow NP \ RelClause \\
RelClause \rightarrow \text{that } NP \text{ ate}
\]

Linguists distinguish between a speaker’s - **competence** (grammatical knowledge) and - **performance** (processing and memory limitations)
CFGs and center embedding

The mouse ate the corn.
The mouse that the snake ate ate the corn.
The mouse that the snake that the hawk ate ate ate the corn.

These sentences are all grammatical. They can be generated by a CFG:

\[
S \rightarrow \text{NP} \quad \text{VP} \\
\text{NP} \rightarrow \text{NP} \quad \text{RelClause} \\
\text{RelClause} \rightarrow \text{that} \quad \text{NP} \quad \text{ate}
\]

Linguists distinguish between a speaker’s
- competence (grammatical knowledge) and
- performance (processing and memory limitations)
Head Words

Each constituent has one word which captures its “essence”.

hat is the “semantic head”

with is the “functional head”

(it is common to choose the functional head)
Head Words

Each constituent has one word which captures its “essence”.

- (S John saw the young boy with the large hat)
Head Words

Each constituent has one word which captures its “essence”.

▶ (S John saw the young boy with the large hat)
Head Words

Each constituent has one word which captures its “essence”.

- (S John **saw** the young boy with the large hat)
- (VP saw the young boy with the large hat)
Head Words

Each constituent has one word which captures its "essence".

- (S John **saw** the young boy with the large hat)
- (VP **saw** the young boy with the large hat)
Head Words

Each constituent has one word which captures its “essence”.

- (S John \textbf{saw} the young boy with the large hat)
- (VP \textbf{saw} the young boy with the large hat)
- (NP the young boy with the large hat)
Head Words

Each constituent has one words which captures its “essence”.

- (S John saw the young boy with the large hat)
- (VP saw the young boy with the large hat)
- (NP the young boy with the large hat)
Head Words

Each constituent has one word which captures its “essence”.

- (S John saw the young boy with the large hat)
- (VP saw the young boy with the large hat)
- (NP the young boy with the large hat)
- (NP the large hat)
Head Words

Each constituent has one word which captures its “essence”.

- (S John **saw** the young boy with the large hat)
- (VP **saw** the young boy with the large hat)
- (NP the young **boy** with the large hat)
- (NP the large **hat**)
Head Words

Each constituent has one word which captures its “essence”.

- (S John **saw** the young boy with the large hat)
- (VP **saw** the young boy with the large hat)
- (NP the young **boy** with the large hat)
- (NP the large **hat**)
- (PP with the large hat)
Each constituent has one word which captures its “essence”.

- (S John \textbf{saw} the young boy with the large hat)
- (VP \textbf{saw} the young boy with the large hat)
- (NP the young \textbf{boy} with the large hat)
- (NP the large \textbf{hat})
- (PP \textbf{with} the large hat)
Head Words

Each constituent has one word which captures its “essence”.

- (S John **saw** the young boy with the large hat)
- (VP **saw** the young boy with the large hat)
- (NP the young **boy** with the large hat)
- (NP the large **hat**)
- (PP **with** the large **hat**)

hat is the “semantic head”
with is the “functional head”
(it is common to choose the functional head)
Head Words

Each constituent has one word which captures its “essence”.

- (S John **saw** the young boy with the large hat)
- (VP **saw** the young boy with the large hat)
- (NP the young **boy** with the large hat)
- (NP the large **hat**)
- (PP **with** the large hat)
  - **hat** is the “semantic head”
  - **with** is the “functional head”
  - (it is common to choose the functional head)
Heads in Context-Free Rules

Add annotations specifying the “head” of each rule:

<table>
<thead>
<tr>
<th>S</th>
<th>NP</th>
<th>VP</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP</td>
<td>Vi</td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>Vt</td>
<td>NP</td>
</tr>
<tr>
<td>VP</td>
<td>VP</td>
<td>PP</td>
</tr>
<tr>
<td>NP</td>
<td>DT</td>
<td>NN</td>
</tr>
<tr>
<td>NP</td>
<td>NP</td>
<td>PP</td>
</tr>
<tr>
<td>PP</td>
<td>IN</td>
<td>NP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vi</th>
<th>sleeps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vt</td>
<td>saw</td>
</tr>
<tr>
<td>NN</td>
<td>man</td>
</tr>
<tr>
<td>NN</td>
<td>woman</td>
</tr>
<tr>
<td>NN</td>
<td>telescope</td>
</tr>
<tr>
<td>DT</td>
<td>the</td>
</tr>
<tr>
<td>IN</td>
<td>with</td>
</tr>
<tr>
<td>IN</td>
<td>in</td>
</tr>
</tbody>
</table>
More about Heads

- Each context-free rule has one “special” child that is the head of the rule. e.g.,
  
  \[
  S \Rightarrow NP \ VP \quad (VP \text{ is the head}) \\
  VP \Rightarrow Vt \ NP \quad (Vt \text{ is the head}) \\
  NP \Rightarrow DT \ NN \ NN \quad (NN \text{ is the head})
  \]

- A core idea in syntax (e.g., see X-bar Theory, Head-Driven Phrase Structure Grammar)

- Some intuitions:
  - The central sub-constituent of each rule.
  - The semantic predicate in each rule.
Rules which Recover Heads: An Example for NPs

If the rule contains NN, NNS, or NNP:
    Choose the rightmost NN, NNS, or NNP

Else If the rule contains an NP: Choose the leftmost NP

Else If the rule contains a JJ: Choose the rightmost JJ

Else If the rule contains a CD: Choose the rightmost CD

Else Choose the rightmost child

e.g.,

NP  ⇒  DT  NNP  NN
NP  ⇒  DT  NN  NNP
NP  ⇒  NP  PP
NP  ⇒  DT  JJ
NP  ⇒  DT
Rules which Recover Heads: An Example for VPs

If the rule contains Vi or Vt: Choose the leftmost Vi or Vt

Else If the rule contains an VP: Choose the leftmost VP

Else Choose the leftmost child

e.g.,

\[
\begin{align*}
\text{VP} & \Rightarrow \text{Vt} \quad \text{NP} \\
\text{VP} & \Rightarrow \text{VP} \quad \text{PP}
\end{align*}
\]
Adding Headwords to Trees

```
S
  NP         VP
  |           |
  DT the     Vt questioned
         NP
           |     |
           DT the NN witness

↓

S(questioned)
  NP(lawyer)   VP(questioned)
     |         |
     DT the   NN(lawyer)       Vt(questioned)
                      |         |
                      the     questioned

                             NP(witness)
                                |     |
                                DT the   NN(witness)
                               |   |
                               the   witness
```
Adding Headwords to Trees (Continued)

A constituent receives its **headword** from its **head child**.

- \( S \Rightarrow NP \quad VP \) (S receives headword from VP)
- \( VP \Rightarrow Vt \quad NP \) (VP receives headword from Vt)
- \( NP \Rightarrow DT \quad NN \) (NP receives headword from NN)
Dependency Representation
Dependency Representation

If we take the head-annotated trees and “forget” about the constituents, we get a representation called “dependency structure”.

Dependency structure capture the relation between words in a sentence.
Dependency Representation

S(questioned)

NP(lawyer)
  DT(the)  NN(lawyer)
    the    lawyer

VP(questioned)
  Vt(questioned)
    questioned

NP(witness)
  DT(the)  NN(witness)
    the    witness

Dependency representation is very common. We will return to it in the future.
Dependency Representation

Dependency representation is very common. We will return to it in the future.
Dependency representation is very common. We will return to it in the future.
Dependency Representation

questioned

lawyer witness

the the
Dependency Representation

questioned

lawyer  witness

the  the

The lawyer questioned the witness
I heard that the lawyer questioned the witness under oath yesterday.
Dependency Representation

I heard that the lawyer questioned the witness under oath yesterday.
Dependency Representations

There are many different dependency representations

- Different choice of heads.
- Different set of labels.
- Each language usually has its own treebank, with own choices.
- A common (and good) one for English: **Stanford Dependencies**
  - Prefer relations between words as heads.
  - About 50 labels.
- Recently, Trees in Stanford-Dependencies available for different languages.
  - Google’s Universal Dependency Treebank
Universal Dependencies

- A multi-national project aiming at producing a consistent set of dependency annotations in many (all!) languages.
Universal Dependencies

- A multi-national project aiming at producing a consistent set of dependency annotations in many (all!) languages.
- Abstract over linguistic differences.
- Same set of parts-of-speech and morphology features.
- Same dependency relations.
- Same choice of heads.
Universal Dependencies

- A multi-national project aiming at producing a consistent set of dependency annotations in many (all!) languages.
- Abstract over linguistic differences.
- Same set of parts-of-speech and morphology features.
- Same dependency relations.
- Same choice of heads.
- Why is this good? why is this interesting?
Universal Dependencies

- A multi-national project aiming at producing a consistent set of dependency annotations in many (all!) languages.
- Abstract over linguistic differences.
- Same set of parts-of-speech and morphology features.
- Same dependency relations.
- Same choice of heads.
- Why is this good? why is this interesting?
- Interesting project/research idea: are the annotations really consistent across languages? do languages differ only in word order?
Let’s analyze!

John saw Mary.
Let’s analyze!

John saw Mary.

a yellow garbage can
Let's analyze!

He said that the boy who was wearing the blue shirt with the white pockets has left the building.
Let’s analyze!

a large pile of carrots and peas was closely guarded by dogs.
Let’s analyze!

They wanted to buy cakes and eat them on the road.
Some tricky cases

I bought soda and pizza for John and Mary.
Some tricky cases

I bought soda and pizza for 4 and 57 cents.
Some tricky cases

I ordered five books but received four.
Some tricky cases

While Sue has many toys, Alice doesn’t have any.
Some tricky cases

Cut, chop and peel the tomatoes.
Some tricky cases

Cut the tomatoes. Put in a bowl.
Coordination is interesting and important.
Missing elements are interesting and important.
  on the border of syntax and discourse.
Lots of work to do!
What did we learn?

• What is a grammar?
• Constituents
  • Constituency tests
• Context Free Grammars
• Some English Constructions
• Head Words
  • Relation to Dependencies
• Dependency Representation