Syntax

• With syntax, we’re moving from labels for discrete items — documents (sentiment analysis), tokens (POS tagging, NER) — to the structure between items.

I shot an elephant in my pajamas
I shot an elephant in my pajamas
Why is syntax important?

```
S
  NP  VP
    I  
  shot NP
    an Nominal
       NP
          PP
            Nominal  in
            elephant  my
            NP  pajamas
```
Why is POS important?

- POS tags are indicative of syntax
- POS = cheap multiword expressions \[(JJ|NN)+ NN\]
- POS tags are indicative of pronunciation ("I contest the ticket" vs "I won the contest")
Why is syntax important?

• Foundation for **semantic analysis** (on many levels of representation: semantic roles, compositional semantics, frame semantics)
Why is syntax important?

• Strong representation for discourse analysis (e.g., coreference resolution)

Bill VBD Jon; he was having a good day.

• Many factors contribute to pronominal coreference (including the specific verb above), but syntactic subjects > objects > objects of prepositions are more likely to be antecedents
Why is syntax important?

Linguistic typology; relative positions of subjects (S), objects (O) and verbs (V)

<table>
<thead>
<tr>
<th></th>
<th>Language</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVO</td>
<td>English, Mandarin</td>
<td>I grabbed the chair</td>
</tr>
<tr>
<td>SOV</td>
<td>Latin, Japanese</td>
<td>I the chair grabbed</td>
</tr>
<tr>
<td>VSO</td>
<td>Hawaiian</td>
<td>Grabbed I the chair</td>
</tr>
<tr>
<td>OSV</td>
<td>Yoda</td>
<td>Patience you must have</td>
</tr>
</tbody>
</table>
"Unfortunately I already had this exact picture tattooed on my chest, but this shirt is very useful in colder weather."

[overlook1977]
Barack Hussein Obama II (born August 4, 1961) is the 44th and current President of the United States, and the first African American to hold the office. Born in Honolulu, Hawaii, Obama is a graduate of Columbia University and Harvard Law School, where he served as president of the Harvard Law Review. He was a community organizer in Chicago before earning his law degree. He worked as a civil rights attorney and taught constitutional law at the University of Chicago Law School between 1992 and 2004.
Syntax

Syntax is fundamentally about the hierarchical structure of language and (in some theories) which sentences are grammatical in a language.

words → phrases → clauses → sentences
Formalisms

Phrase structure grammar
(Chomsky 1957)

Dependency grammar
(Mel’čuk 1988; Tesnière 1959; Pāṇini)
Constituency

- Groups of words ("constituents") behave as single units
- "Behave" = show up in the same distributional environments
everyone likes

a bottle of

makes you drunk

da cocktail with

context
Parts of speech

• Parts of speech are categories of words defined distributionally by the morphological and syntactic contexts a word appears in.
Syntactic distribution

• Substitution test: if a word is replaced by another word, does the sentence remain grammatical?

<table>
<thead>
<tr>
<th>Kim saw the elephant before we did</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
</tr>
<tr>
<td>idea</td>
</tr>
<tr>
<td>*of</td>
</tr>
<tr>
<td>*goes</td>
</tr>
</tbody>
</table>

Bender 2013
<table>
<thead>
<tr>
<th>three parties from Brooklyn</th>
<th>arrive</th>
</tr>
</thead>
<tbody>
<tr>
<td>a high-class spot such as Mindy’s</td>
<td>attracts</td>
</tr>
<tr>
<td>the Broadway coppers</td>
<td>love</td>
</tr>
<tr>
<td>they</td>
<td>sit</td>
</tr>
</tbody>
</table>

Jurafsky and Martin 2017
Syntactic distributions

<table>
<thead>
<tr>
<th>three parties</th>
<th>from Brooklyn</th>
<th>arrive</th>
</tr>
</thead>
<tbody>
<tr>
<td>a high-class</td>
<td>spot such as</td>
<td>Mindy's</td>
</tr>
<tr>
<td>the Broadway</td>
<td>coppers</td>
<td></td>
</tr>
<tr>
<td>they</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

grammatical only when the entire phrase is present, not an individual word in isolation

Jurafsky and Martin 2017
Syntactic distributions

I’d like to fly from Atlanta to Denver on September seventeenth
Formalisms

Phrase structure grammar
(Chomsky 1957)

Dependency grammar
(Mel’čuk 1988; Tesnière 1959; Pāṇini)
A CFG gives a formal way to define what meaningful constituents are and exactly how a constituent is formed out of other constituents (or words). It defines valid structure in a language.
A context-free grammar defines how symbols in a language combine to form valid structures.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>→ Det Nominal</td>
</tr>
<tr>
<td>NP</td>
<td>→ ProperNoun</td>
</tr>
<tr>
<td>Nominal</td>
<td>→ Noun</td>
</tr>
<tr>
<td>Det</td>
<td>→ a</td>
</tr>
<tr>
<td>Noun</td>
<td>→ flight</td>
</tr>
</tbody>
</table>
## Context-free grammar

<table>
<thead>
<tr>
<th></th>
<th>Finite set of non-terminal symbols</th>
<th>NP, VP, S</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td><strong>Finite set of non-terminal symbols</strong></td>
<td>NP, VP, S</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td><strong>Finite alphabet of terminal symbols</strong></td>
<td>the, dog, a</td>
</tr>
<tr>
<td>$R$</td>
<td><strong>Set of production rules, each</strong></td>
<td>$S \rightarrow$ NP VP</td>
</tr>
<tr>
<td></td>
<td>$A \rightarrow \beta$</td>
<td>Noun $\rightarrow$ dog</td>
</tr>
<tr>
<td></td>
<td>$\beta \in (\Sigma, N)$</td>
<td></td>
</tr>
<tr>
<td>$S$</td>
<td><strong>Start symbol</strong></td>
<td></td>
</tr>
</tbody>
</table>
Infinite strings with finite productions

Some sentences go on
Infinite strings with finite productions

- This is the house
- This is the house that Jack built
- This is the cat that lives in the house that Jack built
- This is the dog that chased the cat that lives in the house that Jack built
- This is the flea that bit the dog that chased the cat that lives in the house that Jack built
- This is the virus that infected the flea that bit the dog that chased the cat that lives in the house that Jack built
Derivation

Given a CFG, a derivation is the sequence of productions used to generate a string of words (e.g., a sentence), often visualized as a parse tree.
Language

The formal language defined by a CFG is the set of strings derivable from \( S \) (start symbol)
Figure 11.2  The lexicon for $L_0$.

<table>
<thead>
<tr>
<th>Grammar Rules</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow NP \ VP$</td>
<td>I + want a morning flight</td>
</tr>
<tr>
<td>$NP \rightarrow$ Pronoun</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>Proper-Noun</td>
</tr>
<tr>
<td></td>
<td>Det Nominal</td>
</tr>
<tr>
<td>Nominal $\rightarrow$ Nominal Noun</td>
<td>morning + flight</td>
</tr>
<tr>
<td></td>
<td>Noun</td>
</tr>
<tr>
<td>$VP \rightarrow$ Verb</td>
<td>do</td>
</tr>
<tr>
<td></td>
<td>Verb NP</td>
</tr>
<tr>
<td></td>
<td>Verb NP PP</td>
</tr>
<tr>
<td></td>
<td>Verb PP</td>
</tr>
<tr>
<td>$PP \rightarrow$ Preposition NP</td>
<td>from + Los Angeles</td>
</tr>
</tbody>
</table>

Figure 11.3  The grammar for $L_0$, with example phrases for each rule.
Bracketed notation

NP
  \[\text{Det the}\]
  \[\text{Nominal [Noun flight]}\]
Constituents

Every internal node is a phrase

- my pajamas
- in my pajamas
- elephant in my pajamas
- an elephant in my pajamas
- shot an elephant in my pajamas
- I shot an elephant in my pajamas

Each phrase could be replaced by another of the same type of constituent
<table>
<thead>
<tr>
<th>Production</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow VP$</td>
<td>Imperatives</td>
<td>“Show me the right way”</td>
</tr>
<tr>
<td>$S \rightarrow NP \ VP$</td>
<td>Declaratives</td>
<td>“The dog barks”</td>
</tr>
<tr>
<td>$S \rightarrow Aux \ NP \ VP$</td>
<td>Yes/no questions</td>
<td>“Will you show me the right way?”</td>
</tr>
<tr>
<td>$S \rightarrow Wh-NP \ VP$</td>
<td>Wh-subject question</td>
<td>“Which flights serve breakfast?”</td>
</tr>
</tbody>
</table>
Verb phrases

\[
\begin{array}{|c|c|c|}
\hline
\text{VP} & \rightarrow & \text{Verb} \\
\hline
\text{VP} & \rightarrow & \text{Verb NP} \\
\hline
\text{VP} & \rightarrow & \text{Verb NP PP} \\
\hline
\text{VP} & \rightarrow & \text{Verb PP} \\
\hline
\text{VP} & \rightarrow & \text{Verb S} \\
\hline
\text{VP} & \rightarrow & \text{Verb VP} \\
\hline
\end{array}
\]

- disappear
- prefer a morning flight
- prefer a morning flight on Tuesday
- leave on Tuesday
- I think [s I want a new flight]
- want [\text{vp} to fly today]

Not every verb can appear in each of these productions.
Verb phrases

<table>
<thead>
<tr>
<th>Production</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP → Verb</td>
<td>*I filled</td>
</tr>
<tr>
<td>VP → Verb NP</td>
<td>*I exist the morning flight</td>
</tr>
<tr>
<td>VP → Verb NP PP</td>
<td>*I exist the morning flight on Tuesday</td>
</tr>
<tr>
<td>VP → Verb PP</td>
<td>*I filled on Tuesday</td>
</tr>
<tr>
<td>VP → Verb S</td>
<td>*I exist [S I want a new flight]</td>
</tr>
<tr>
<td>VP → Verb VP</td>
<td>*I fill [VP to fly today]</td>
</tr>
</tbody>
</table>

Not every verb can appear in each of these productions
Subcategorization

- Verbs are compatible with different complements
  - Transitive verbs take direct object NP ("I filled the tank")
  - Intransitive verbs don’t ("I exist")
Subcategorization

- The set of possible complements of a verb is its subcategorization frame.

<table>
<thead>
<tr>
<th>VP</th>
<th>→</th>
<th>Verb VP</th>
<th>* I fill [VP to fly today]</th>
</tr>
</thead>
<tbody>
<tr>
<td>VP</td>
<td>→</td>
<td>Verb VP</td>
<td>I want [VP to fly today]</td>
</tr>
</tbody>
</table>
### Coordination

Coordination here also helps us establish whether a group of words forms a constituent.

<table>
<thead>
<tr>
<th>NP</th>
<th>NP and NP</th>
<th>the dogs and the cats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>Nominal and Nominal</td>
<td>dogs and cats</td>
</tr>
<tr>
<td>VP</td>
<td>VP and VP</td>
<td>I came and saw and conquered</td>
</tr>
<tr>
<td>JJ</td>
<td>JJ and JJ</td>
<td>beautiful and red</td>
</tr>
<tr>
<td>S</td>
<td>S and S</td>
<td>I came and I saw and I conquered</td>
</tr>
</tbody>
</table>
I shot an elephant in my pajamas
S
  NP
    I
      shot
  VP
    NP
      an
      Nominal
        Nominal
          PP
            elephant
            in
            NP
              my
              pajamas

S
  NP
    I
      VP
      PP
        shot
        NP
          an
          Nominal
            elephant
            in
            NP
              my
              pajamas
Parseval (1991): Represent each tree as a collection of tuples:

\(<l_1, i_1, j_1>, \ldots, <l_n, i_n, j_n>\)

- \(l_k\) = label for \(k\)th phrase
- \(i_k\) = index for first word in \(k\)th phrase
- \(j_k\) = index for last word in \(k\)th phrase
Evaluation

I_{1} shot_{2} an_{3} elephant_{4} in_{5} my_{6} pajamas_{7}

- <S, 1, 7>
- <NP, 1, 1>
- <VP, 2, 7>
- <VP, 2, 4>
- <NP, 3, 4>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>
S
  NP
    shot
  VP
    an
    Nominal
      Nominal
      PP
      elephant
      in
      NP
      my
      pajamas

S
  NP
    shot
  VP
    an
    Nominal
      elephant
    PP
      in
      NP
      my
      pajamas
Evaluation

I shot an elephant in my pajamas

- <S, 1, 7>
- <NP, 1, 1>
- <VP, 2, 7>
- <VP, 2, 4>
- <NP, 3, 4>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>
- <S, 1, 7>
- <NP, 1, 1>
- <VP, 2, 7>
- <NP, 3, 7>
- <Nominal, 4, 7>
- <Nominal, 4, 4>
- <PP, 5, 7>
- <NP, 6, 7>
Evaluation

Calculate precision, recall, F1 from these collections of tuples

• Precision: number of tuples in tree 1 also in tree 2, divided by number of tuples in tree 1

• Recall: number of tuples in tree 1 also in tree 2, divided by number of tuples in tree 2
Evaluation

I_{1} shot_{2} an_{3} elephant_{4} in_{5} my_{6} pajamas_{7}
CFGs

- Building a CFG by hand is really hard
- To capture all (and only) grammatical sentences, need to exponentially increase the number of categories (e.g., detailed subcategorization info)

<table>
<thead>
<tr>
<th>Category</th>
<th>Rule</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb-with-no-complement</td>
<td>→ disappear</td>
<td></td>
</tr>
<tr>
<td>Verb-with-S-complement</td>
<td>→ said</td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>→ Verb-with-no-complement</td>
<td></td>
</tr>
<tr>
<td>VP</td>
<td>→ Verb-with-S-complement S</td>
<td></td>
</tr>
</tbody>
</table>
**CFGs**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb-with-no-complement</td>
<td>→ disappear</td>
</tr>
<tr>
<td>Verb-with-S-complement</td>
<td>→ said</td>
</tr>
<tr>
<td>VP</td>
<td>→ Verb-with-no-complement</td>
</tr>
<tr>
<td>VP</td>
<td>→ Verb-with-S-complement S</td>
</tr>
</tbody>
</table>

- disappear
- said he is going to the airport
- *disappear he is going to the airport
Treebanks

• Rather than create the rules by hand, we can annotate sentences with their syntactic structure and then extract the rules from the annotations

• Treebanks: collections of sentences annotated with syntactic structure
Penn Treebank

Example rules extracted from this single annotation

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>NNP NNP</td>
</tr>
<tr>
<td>NP-SBJ</td>
<td>NP, ADJP</td>
</tr>
<tr>
<td>S</td>
<td>NP-SBJ VP</td>
</tr>
<tr>
<td>VP</td>
<td>VB NP PP-CLR NP-TMP</td>
</tr>
</tbody>
</table>
Penn Treebank

NP → DT JJ NN
NP → DT JJ NNS
NP → DT JJ NN NN
NP → DT JJ JJ NN
NP → DT JJ CD NNS
NP → RB DT JJ NN NN
NP → RB DT JJ JJ NNS
NP → DT JJ JJ NNP NNS
NP → DT NNP NNP NNP NNP JJ NN
NP → DT JJ NNP CC JJ JJ NN NNS
NP → RB DT JJS NN NN SBAR
NP → DT VBG JJ NNP NNP CC NNP
NP → DT JJ NNS , NNS CC NN NNS NN
NP → DT JJ JJ VBG NN NNP NNP FW NNP
NP → NP JJ , JJ ‘‘ SBAR ’’ NNS
A basic CFG allows us to check whether a sentence is grammatical in the language it defines.

Binary decision: a sentence is either in the language (a series of productions yields the words we see) or it is not.

Where would this be useful?
• Probabilistic context-free grammar: each production is also associated with a probability.

• This lets us calculate the probability of a parse for a given sentence; for a given parse tree $T$ for sentence $S$ comprised of $n$ rules from $R$ (each $A \rightarrow \beta$):

\[
P(T, S) = \prod_{i}^{n} P(\beta \mid A)
\]
# PCFG

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Finite set of non-terminal symbols</td>
<td>NP, VP, S</td>
</tr>
<tr>
<td>$\Sigma$</td>
<td>Finite alphabet of terminal symbols</td>
<td>the, dog, a</td>
</tr>
<tr>
<td>$R$</td>
<td>Set of production rules, each $A \rightarrow \beta [p]$</td>
<td>$S \rightarrow$ NP VP</td>
</tr>
<tr>
<td></td>
<td>$p = P(\beta \mid A)$</td>
<td>Noun $\rightarrow$ dog</td>
</tr>
<tr>
<td>$S$</td>
<td>Start symbol</td>
<td></td>
</tr>
</tbody>
</table>
PCFG

\[ \sum_{\beta} P(A \rightarrow \beta) = 1 \]

(equivalently)

\[ \sum_{\beta} P(\beta \mid A) = 1 \]
Estimating PCFGs

How do we calculate $P(A \rightarrow \beta)$?
Estimating PCFGs

$$\sum_\beta P(\beta \mid A) = \frac{C(A \rightarrow \beta)}{\sum_\gamma C(A \rightarrow \gamma)}$$

(equivalently)

$$\sum_\beta P(\beta \mid A) = \frac{C(A \rightarrow \beta)}{C(A)}$$
| A             | β             | P(β | NP) |
|---------------|---------------|-------|
| NP → NP PP    |               | 0.092 |
| NP → DT NN    |               | 0.087 |
| NP → NN       |               | 0.047 |
| NP → NNS      |               | 0.042 |
| NP → DT JJ NN |               | 0.035 |
| NP → NNP      |               | 0.034 |
| NP → NNP NNP  |               | 0.029 |
| NP → JJ NNS   |               | 0.027 |
| NP → QP -NONE-|               | 0.018 |
| NP → NP SBAR  |               | 0.017 |
| NP → NP PP-LOC|               | 0.017 |
| NP → JJ NN    |               | 0.015 |
| NP → DT NNS   |               | 0.014 |
| NP → CD       |               | 0.014 |
| NP → NN NNS   |               | 0.013 |
| NP → DT NN NN |               | 0.013 |
| NP → NP CC NP |               | 0.013 |
PCFGs

- A CFG tells us whether a sentence is in the language it defines.
- A PCFG gives us a mechanism for assigning scores (here, probabilities) to different parses for the same sentence.
$P(\text{NP VP} \mid \text{S})$
\[
P(NP \ VP | S) \
\times P(\text{Nominal} | NP)
\]
$P(\text{NP VP} \mid \text{S})$
\times P(\text{Nominal} \mid \text{NP})
\times P(\text{Pronoun} \mid \text{Nominal})$
\[
P(NP \text{ VP} | S)
\times P(\text{Nominal} | NP)
\times P(\text{Pronoun} | \text{Nominal})
\times P(I | \text{Pronoun})
\]
\[
P(NP \ VP \ | \ S) \\
\times P(\text{Nominal} \ | \ NP) \\
\times P(\text{Pronoun} \ | \ \text{Nominal}) \\
\times P(\text{I} \ | \ \text{Pronoun}) \\
\times P(\text{VP PP} \ | \ \text{VP})
\]
$P(\text{NP VP | S})$
$\times P(\text{Nominal | NP})$
$\times P(\text{Pronoun | Nominal})$
$\times P(\text{I | Pronoun})$
$\times P(\text{VP PP | VP})$
$\times P(\text{Verb NP | VP})$
\[
P(NP \ VP | S) \\
\times P(\text{Nominal} | NP) \\
\times P(\text{Pronoun} | \text{Nominal}) \\
\times P(I | \text{Pronoun}) \\
\times P(\text{VP PP} | \text{VP}) \\
\times P(\text{Verb NP} | \text{VP}) \\
\times P(\text{shot} | \text{Verb})
\]
\[
P(\text{NP VP } | \text{ S}) \\
\times P(\text{Nominal } | \text{ NP}) \\
\times P(\text{Pronoun } | \text{ Nominal}) \\
\times P(\text{I } | \text{ Pronoun}) \\
\times P(\text{VP PP } | \text{ VP}) \\
\times P(\text{Verb NP } | \text{ VP}) \\
\times P(\text{shot } | \text{ Verb}) \\
\times P(\text{Det Nominal } | \text{ NP})
\]
\[ P(NP \ VP \ |\ S) \]
\[ \times P(\text{Nominal} \ |\ NP) \]
\[ \times P(\text{Pronoun} \ |\ \text{Nominal}) \]
\[ \times P(I \ |\ \text{Pronoun}) \]
\[ \times P(\text{VP} \ PP \ |\ \text{VP}) \]
\[ \times P(\text{Verb} \ NP \ |\ \text{VP}) \]
\[ \times P(\text{shot} \ |\ \text{Verb}) \]
\[ \times P(\text{Det Nominal} \ |\ NP) \]
\[ \times P(\text{an} \ |\ \text{Det}) \]
\[ S \rightarrow NP \mid VP \]
\[ NP \rightarrow Nominal \mid Pronoun \]
\[ VP \rightarrow Verb \mid NP \]
\[ PP \]
\[ Verb \rightarrow shot \]
\[ NP \rightarrow Det \mid Nominal \]
\[ Det \rightarrow an \]
\[ Nominal \rightarrow Noun \]
P(NP VP | S)
×P(Nominal | NP)
×P(Pronoun | Nominal)
×P(I | Pronoun)
×P(VP PP | VP)
×P(Verb NP | VP)
×P(shot | Verb)
×P(Det Nominal | NP)
×P(an | Det)
×P(Noun | Nominal)
×P(elephant | Noun)
\[ P(T, S) = \prod_{i=1}^{n} P(\beta | A) \]
PCFGs

- A PCFG gives us a mechanism for assigning scores (here, probabilities) to different parses for the same sentence.

- But we often care about is finding the single best parse with the highest probability.