Context free grammar

\[ S \rightarrow NP \ VP \]

left hand side (single symbol) right hand side (one or more symbols)

Formally...

\[ G = (NT, T, P, S) \]

- \( NT \): finite set of nonterminal symbols
- \( T \): finite set of terminal symbols, \( NT \) and \( T \) are disjoint
- \( P \): finite set of productions of the form \( A \rightarrow \alpha, \ A \in NT \) and \( \alpha \in (T \cup NT)^* \)
- \( S \in NT \): start symbol

Assignment 3 out: due next Monday

Quiz #1
Many possible CFGs for English, here is an example (fragment):

- **S → NP VP**
- **VP → V NP**
- **NP → DetP N | DetP AdjP N**
- **AdjP → Adj | Adv AdjP**
- **N → boy | girl**
- **V → sees | likes**
- **Adj → big | small**
- **Adv → very**
- **DetP → a | the**

**Derivations in a CFG**

- **S → NP VP**
- **VP → V NP**
- **NP → DetP N | DetP AdjP N**
- **AdjP → Adj | Adv AdjP**
- **N → boy | girl**
- **V → sees | likes**
- **Adj → big | small**
- **Adv → very**
- **DetP → a | the**

**What can we do?**
Derivations in a CFG

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the

S → NP VP
VP → V NP
NP → DetP N | DetP AdjP N
AdjP → Adj | Adv AdjP
N → boy | girl
V → sees | likes
Adj → big | small
Adv → very
DetP → a | the
Derivations in a CFG

\[ S \rightarrow NP \; VP \]
\[ VP \rightarrow V \; NP \]
\[ NP \rightarrow DetP \; N \mid DetP \; AdjP \; N \]
\[ AdjP \rightarrow Adj \mid Adv \; AdjP \]
\[ N \rightarrow \text{boy} \mid \text{girl} \]
\[ V \rightarrow \text{sees} \mid \text{likes} \]
\[ Adv \rightarrow \text{very} \]
\[ DetP \rightarrow a \mid \text{the} \]

the boy likes NP

Derivations of CFGs

String rewriting system: we derive a string

Derivation history shows constituent tree:

the boy likes a girl
Parsing

Parsing is the field of NLP interested in automatically determining the syntactic structure of a sentence.

Parsing can be thought of as determining what sentences are "valid" English sentences.

As a byproduct, we often can get the structure.

Given a CFG and a sentence, determine the possible parse tree(s)

```
S -> NP VP
NP -> N
NP -> PRP
NP -> N PP
VP -> V NP
VP -> V NP PP
PP -> IN N
PRP -> I
V -> eat
N -> sushi
N -> tuna
IN -> with
```

What parse trees are possible for this sentence?

How did you do it?

What if the grammar is much larger?

What is the difference between these parses?

How can we decide between these?
A Simple PCFG

Probabilities!

```
S  →  NP  VP  1.0
VP  →  V   NP  0.7
VP  →  VP  PP  0.3
PP  →  P   NP  1.0
P   →  with 1.0
V   →  saw  1.0
NP  →  astronomers  0.1
NP  →  ears  0.18
NP  →  stars  0.04
PP  →  with  ears  0.1
V   →  saw  stars  with ears  0.1
```

Just like n-gram language modeling, PCFGs break the sentence generation process into smaller steps/probabilities.

The probability of a parse is the product of the PCFG rules.

What are the different interpretations here?

Which do you think is more likely?
Parsing problems
Pick a model
- e.g. CFG, PCFG, …

Train (or learn) a model
- What CFG/PCFG rules should I use?
- Parameters (e.g. PCFG probabilities)?
- What kind of data do we have?

Parsing
- Determine the parse tree(s) given a sentence

PCFG: Training
If we have example parsed sentences, how can we learn a set of PCFGs?

Tree Bank

Supervised PCFG Training

English

Extracting the rules

I eat sushi with tuna

What CFG rules occur in this tree?

Estimating PCFG Probabilities

We can extract the rules from the trees

How do we go from the extracted CFG rules to PCFG rules?
Estimating PCFG Probabilities

Extract the rules from the trees

Calculate the probabilities using MLE

\[ P(\alpha \rightarrow \beta | \alpha) = \frac{\text{count}(\alpha \rightarrow \beta)}{\sum \text{count}(\alpha \rightarrow \gamma)} \]

Grammar Equivalence

What does it mean for two grammars to be equal?

Grammar Equivalence

Weak equivalence: grammars generate the same set of strings
- Grammar 1: \( NP \rightarrow \text{DetP} N \) and \( \text{DetP} \rightarrow \alpha \mid \text{the} \)
- Grammar 2: \( NP \rightarrow \alpha N \mid \text{the} \ N \)

Strong equivalence: grammars have the same set of derivation trees
- With CFGs, possible only with useless rules
- Grammar 2: \( NP \rightarrow \alpha N \mid \text{the} \ N \)
- Grammar 3: \( NP \rightarrow \alpha N \mid \text{the} N, \text{DetP} \rightarrow \text{many} \)

Estimating PCFG Probabilities

Occurrences

<table>
<thead>
<tr>
<th>Rule</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP )</td>
<td>10</td>
</tr>
<tr>
<td>( S \rightarrow V NP )</td>
<td>3</td>
</tr>
<tr>
<td>( S \rightarrow VP PP )</td>
<td>2</td>
</tr>
<tr>
<td>( NP \rightarrow N )</td>
<td>7</td>
</tr>
<tr>
<td>( NP \rightarrow N PP )</td>
<td>3</td>
</tr>
<tr>
<td>( NP \rightarrow DT N )</td>
<td>6</td>
</tr>
</tbody>
</table>

\[ P( S \rightarrow V NP) = \frac{\text{count}(S \rightarrow V NP)}{\text{count}(S)} = \frac{3}{15} \]
Normal Forms

There are weakly equivalent normal forms (Chomsky Normal Form, Greibach Normal Form)

A CFG is in Chomsky Normal Form (CNF) if all productions are of one of two forms:
- \( A \rightarrow BC \) with \( A, B, C \) nonterminals
- \( A \rightarrow \alpha \), with \( A \) a nonterminal and \( \alpha \) a terminal

Every CFG has a weakly equivalent CFG in CNF

Probabilistic Grammar Conversion

<table>
<thead>
<tr>
<th>Original Grammar</th>
<th>Chomsky Normal Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \rightarrow NP \ VP ) 0.8</td>
<td>( S \rightarrow NP \ VP ) 0.8</td>
</tr>
<tr>
<td>( S \rightarrow Aux \ NP \ VP ) 0.1</td>
<td>( S \rightarrow Aux \ VP ) 0.1</td>
</tr>
<tr>
<td>( S \rightarrow NP ) 0.1</td>
<td>( S \rightarrow \lambda ) 1.0</td>
</tr>
<tr>
<td>( NP \rightarrow Pronoun ) 0.2</td>
<td>( NP \rightarrow I ) 0.1</td>
</tr>
<tr>
<td>( NP \rightarrow Proper-Noun ) 0.2</td>
<td>( NP \rightarrow he ) 0.02</td>
</tr>
<tr>
<td>( NP \rightarrow Det \ Nominal ) 0.6</td>
<td>( NP \rightarrow she ) 0.02</td>
</tr>
<tr>
<td>( Nominal \rightarrow Noun ) 0.3</td>
<td>( NP \rightarrow I ) 0.04</td>
</tr>
<tr>
<td>( Nominal \rightarrow Det \ Nominal ) 0.6</td>
<td>( NP \rightarrow he ) 0.06</td>
</tr>
<tr>
<td>( Nominal \rightarrow Nominal Noun ) 0.2</td>
<td>( Nominal \rightarrow Nominal Noun ) 0.03</td>
</tr>
<tr>
<td>( Nominal \rightarrow Nominal PP ) 0.5</td>
<td>( Nominal \rightarrow Nominal PP ) 0.15</td>
</tr>
<tr>
<td>( VP \rightarrow Verb ) 0.2</td>
<td>( VP \rightarrow book ) 0.06</td>
</tr>
<tr>
<td>( VP \rightarrow VP ) 0.5</td>
<td>( VP \rightarrow include ) 0.01</td>
</tr>
<tr>
<td>( VP \rightarrow VP PP ) 0.3</td>
<td>( VP \rightarrow prefer ) 0.06</td>
</tr>
<tr>
<td>( PP \rightarrow Prep ) 1.0</td>
<td></td>
</tr>
</tbody>
</table>
## Grammar questions

<table>
<thead>
<tr>
<th>Can we determine if a sentence is grammatical?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a sentence, can we determine the syntactic structure?</td>
</tr>
<tr>
<td>Can we determine how likely a sentence is to be grammatical? to be an English sentence?</td>
</tr>
<tr>
<td>Can we generate candidate, grammatical sentences?</td>
</tr>
</tbody>
</table>

Next time: parsing