Simplified View of Linguistics

Phonetics: “The study of pronunciation of words”

Nikolai Trubetzkoy in Grundzüge der Phonologie (1939) defines phonology as “the study of sound pertaining to the system of language,” as opposed to phonetics, which is “the study of sound pertaining to the act of speech.”

http://en.wikipedia.org/wiki/Phonology

Not to be confused with…

Phonology: “The areas of linguistics that describes the systematic way that sounds are differently realized in different environments”

—The book
Context free grammar

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 \( A \to \alpha \), \( A \in NT \) and \( \alpha \in (T \cup NT)^* \)
- \( S \in NT \): start symbol

CFG: Example

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

- \( S \to NP \ VP \)
- \( VP \to V NP \)
- \( NP \to DetP N \mid AdjP NP \)
- \( AdjP \to Adj \mid Adv AdjP \)
- \( N \to boy \mid girl \)
- \( V \to sees \mid likes \)
- \( Adj \to big \mid small \)
- \( Adv \to very \)
- \( DetP \to a \mid the \)

Grammar questions

- Can we determine if a sentence is grammatical?
- Given a sentence, can we determine the syntactic structure?
- Can we determine how likely a sentence is to be grammatical? to be an English sentence?
- Can we generate candidate, grammatical sentences?

Which of these can we answer with a CFG? How?
Grammar questions

Can we determine if a sentence is grammatical?
- Is it accepted/recognized by the grammar
- Applying rules right to left, do we get the start symbol?

Given a sentence, can we determine the syntactic structure?
- Keep track of the rules applied...

Can we determine how likely a sentence is to be grammatical? to be an English sentence?
- Not yet... no notion of "likelihood" (probability)

Can we generate candidate, grammatical sentences?
- Start from the start symbol, randomly pick rules that apply (i.e. left hand side matches)

Derivations in a CFG

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

$S \rightarrow \text{NP } \text{VP}$
$\text{VP} \rightarrow \text{V } \text{NP}$
$\text{NP} \rightarrow \text{DetP } \text{N} | \text{AdjP } \text{NP}$
$\text{AdjP} \rightarrow \text{Adj} | \text{Adv AdjP}$
$\text{N} \rightarrow \text{boy } | \text{girl}$
$\text{V} \rightarrow \text{sees } | \text{likes}$
$\text{Adj} \rightarrow \text{big } | \text{small}$
$\text{Adv} \rightarrow \text{very}$
$\text{DetP} \rightarrow \text{a } | \text{the}$

Order of Derivation Irrelevant

Derivations in a CFG

$S \rightarrow \text{NP } \text{VP}$
$\text{VP} \rightarrow \text{V } \text{NP}$
$\text{NP} \rightarrow \text{DetP } \text{N} | \text{AdjP } \text{NP}$
$\text{AdjP} \rightarrow \text{Adj} | \text{Adv AdjP}$
$\text{N} \rightarrow \text{boy } | \text{girl}$
$\text{V} \rightarrow \text{sees } | \text{likes}$
$\text{Adj} \rightarrow \text{big } | \text{small}$
$\text{Adv} \rightarrow \text{very}$
$\text{DetP} \rightarrow \text{a } | \text{the}$

String rewriting system: we derive a string

Derivation history shows constituent tree:

the boy likes a girl

the boy likes a girl

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 by product, we often can get the structure.

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

\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow N \\
NP & \rightarrow PRP \\
NP & \rightarrow N PP \\
VP & \rightarrow V NP \\
VP & \rightarrow V NP PP \\
PP & \rightarrow IN N \\
PRP & \rightarrow I \\
V & \rightarrow eat \\
N & \rightarrow sushi \\
N & \rightarrow tuna \\
IN & \rightarrow with
\end{align*}

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  
NP → NP PP 0.4  
VP → V NP 0.7  
NP → astronomers 0.1  
VP → VP PP 0.3  
NP → ears 0.18  
P → with 1.0  
NP → stars 0.18  
V → saw 1.0  
NP → telescope 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

Extracting the rules

- 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

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

<table>
<thead>
<tr>
<th>Production</th>
<th>Count</th>
<th>Occurrences</th>
</tr>
</thead>
<tbody>
<tr>
<td>S → NP VP</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>S → V NP</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>S → VP PP</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>NP → N</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>NP → N PP</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>NP → DT N</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

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

Grammar Equivalence

- Weak equivalence: grammars generate same set of strings
  - Grammar 1: NP → DetP N and DetP → a | the
  - Grammar 2: NP → a N | the N

- Strong equivalence: grammars have same set of derivation trees
  - With CFGs, possible only with useless rules
  - Grammar 2: NP → a N | the N
  - Grammar 3: NP → a N | the N, DetP → many
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:

- \( A \to BC \) with \( A, B, C \) nonterminals
- \( A \to a \) with \( A \) a nonterminal and \( a \) a terminal

Every CFG has a weakly equivalent CFG in CNF

CNF Grammar

<table>
<thead>
<tr>
<th>Production</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \to VP )</td>
<td>0.8</td>
</tr>
<tr>
<td>( VP \to VB NP )</td>
<td>0.1</td>
</tr>
<tr>
<td>( VP \to VP PP )</td>
<td>0.1</td>
</tr>
<tr>
<td>( NP \to DT NN )</td>
<td>0.6</td>
</tr>
<tr>
<td>( NP \to NN )</td>
<td>0.3</td>
</tr>
<tr>
<td>( PP \to IN NP )</td>
<td>1.0</td>
</tr>
<tr>
<td>( DT \to ) the</td>
<td>0.8</td>
</tr>
<tr>
<td>( IN \to with )</td>
<td>0.1</td>
</tr>
<tr>
<td>( VB \to film )</td>
<td>0.5</td>
</tr>
<tr>
<td>( VB \to trust )</td>
<td>0.5</td>
</tr>
<tr>
<td>( NN \to man )</td>
<td>0.2</td>
</tr>
<tr>
<td>( NN \to film )</td>
<td>0.2</td>
</tr>
<tr>
<td>( NN \to trust )</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Probabilistic Grammar Conversion

<table>
<thead>
<tr>
<th>Original Grammar</th>
<th>Chomsky Normal Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S \to NP VP )</td>
<td>0.8</td>
</tr>
<tr>
<td>( NP \to Pronoun )</td>
<td>0.1</td>
</tr>
<tr>
<td>( NP \to Proper-Noun )</td>
<td>0.2</td>
</tr>
<tr>
<td>( NP \to Det-Nominal )</td>
<td>0.6</td>
</tr>
<tr>
<td>( Nominal \to Noun )</td>
<td>0.3</td>
</tr>
<tr>
<td>( VP \to Verb )</td>
<td>0.2</td>
</tr>
<tr>
<td>( VP \to VP PP )</td>
<td>0.3</td>
</tr>
<tr>
<td>( PP \to Prep-NP )</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the capital of this state?</td>
</tr>
</tbody>
</table>
Grammar questions

Can we determine if a sentence is grammatical?

Given a sentence, can we determine the syntactic structure?

Can we determine how likely a sentence is to be grammatical to be an English sentence?

Can we generate candidate, grammatical sentences?

Next time: parsing