Lecture 8:
What’s the matter with CFGs?
How can we learn from our mistakes?

Professor Robert C. Berwick
berwick@csail.mit.edu
Menu

• What’s wrong with CFGs?
• Featuritis – how do we know which features to use? Ans: we will see how to learn them (in a bit)
• The Slash ‘N Burn Strategy: fillers & gaps
• What’s wrong with PCFGs?
• Learning parameters for language models: the EM method
  • First: cone heads
  • Hot today – hot tamale
Agreement features example 1

% start S
S -> NP[NUM=?n] VP[NUM=?n]

# NP expansion productions
NP[NUM=?n] -> N[NUM=?n]
NP[NUM=?n] -> PropN[NUM=?n]
NP[NUM=pl] -> N[NUM=pl]

# VP expansion productions

Det[NUM=sg] -> 'this' | 'every'
Det[NUM=pl] -> 'these' | 'all'
Det -> 'the' | 'some' | 'several'
PropN[NUM=sg] -> 'Kim' | 'Jody'
N[NUM=sg] -> 'dog' | 'girl' | 'car' | 'child'
N[NUM=pl] -> 'dogs' | 'girls' | 'cars' | 'children'
IV[TENSE=pres, NUM=sg] -> 'disappears' | 'walks'
TV[TENSE=pres, NUM=sg] -> 'sees' | 'likes'
Parse *Kim likes children*
Trace showing bottom-up combination & feature checking as each left-hand nonterminal phrase is built

Feature Bottom Up Predict Combine Rule:
|[-----]    .    .| [0:1] PropN[NUM='sg'] -> 'Kim' *
Feature Bottom Up Predict Combine Rule:
|[-----]    .    .| [0:1] NP[NUM='sg'] -> PropN[NUM='sg'] *
Feature Bottom Up Predict Combine Rule:
Feature Bottom Up Predict Combine Rule:
|.    [----]    .| [1:2] TV[NUM='sg', TENSE='pres'] -> 'likes' *
Feature Bottom Up Predict Combine Rule:
                      NP[] {?n: 'sg', ?t: 'pres'}
Feature Bottom Up Predict Combine Rule:
|.    .    [----]| [2:3] N[NUM='pl'] -> 'children' *

```
NP[Num= sg]                     VP[Num=         ]
        |                                           |
PropN[Num = sg ]                TV[Num=sg]
        |                                |         |
Kim  [Num=sg] likes  [Num=sg, Tense-pres]
```
End of trace: NP, VP, S checked

Feature Bottom Up Predict Combine Rule:
| . . [----]  [2:3] NP[NUM='pl'] -> N[NUM='pl'] *

Feature Bottom Up Predict Combine Rule:
{?n: 'pl'}

Feature Single Edge Fundamental Rule:
| . [----------]  [1:3] VP[NUM='sg', TENSE='pres'] -> TV[NUM='sg', TENSE='pres'] NP[] *

Feature Single Edge Fundamental Rule:
|[=============]  [0:3] S[] -> NP[NUM='sg'] VP[NUM='sg'] *
More challenging: German

<table>
<thead>
<tr>
<th>Case</th>
<th>Masc</th>
<th>Fem</th>
<th>Neut</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nom</td>
<td>der</td>
<td>die</td>
<td>das</td>
<td>die</td>
</tr>
<tr>
<td>Gen</td>
<td>des</td>
<td>der</td>
<td>des</td>
<td>der</td>
</tr>
<tr>
<td>Dat</td>
<td>dem</td>
<td>der</td>
<td>dem</td>
<td>den</td>
</tr>
<tr>
<td>Acc</td>
<td>den</td>
<td>die</td>
<td>das</td>
<td>die</td>
</tr>
</tbody>
</table>

Morphological Paradigm for the German definite Article

a. Die Katze sieht den Hund
   the.NOM.FEM.SG cat.3.FEM.SG see.3.SG the.ACC.MASC.SG dog.3.MASC.SG
   ‘the cat sees the dog’

b. *Die Katze sieht dem Hund
   the.NOM.FEM.SG cat.3.FEM.SG see.3.SG dog.3.MASC.SG
   ‘the cat sees the dog’

c. Die Katze hilft dem Hund
   the.NOM.FEM.SG cat.3.FEM.SG help.3.SG the.DAT.MASC.SG dog.3.MASC.SG
   ‘the cat helps the dog’

d. *Die Katze hilft den Hund
   the.NOM.FEM.SG cat.3.FEM.SG help.3.SG the.ACC.MASC.SG dog.3.MASC.SG
   ‘the cat helps the dog’
‘Simple’ agreement grammar for German - not complex agr feature

# Singular determiners

# masc
Det[CASE=nom, AGR=[GND=masc, PER=3, NUM=sg]] -> 'der'
Det[CASE=dat, AGR=[GND=masc, PER=3, NUM=sg]] -> 'dem'
Det[CASE=acc, AGR=[GND=masc, PER=3, NUM=sg]] -> 'den'
...

# fem
Det[CASE=nom, AGR=[GND=fem, PER=3, NUM=sg]] -> 'die'
Det[CASE=dat, AGR=[GND=fem, PER=3, NUM=sg]] -> 'der'
Det[CASE=acc, AGR=[GND=fem, PER=3, NUM=sg]] -> 'die'

# Plural determiners
Det[CASE=nom, AGR=[PER=3, NUM=pl]] -> 'die
...

Nouns
N[AGR=[GND=fem, PER=3, NUM=sg]] -> 'Katze'
N[AGR=[GND=fem, PER=3, NUM=pl]] -> 'Katzen'
Noun-Verb agreement in ‘case’

\[ S \rightarrow \text{NP}[\text{CASE}=\text{nom}, \text{AGR}=?a] \ \text{VP}[\text{AGR}=?a] \]
\[ \text{NP}[\text{CASE}=?c, \text{AGR}=?a] \rightarrow \text{PRO}[\text{CASE}=?c, \text{AGR}=?a] \]
\[ \text{NP}[\text{CASE}=?c, \text{AGR}=?a] \rightarrow \text{Det}[\text{CASE}=?c, \text{AGR}=?a] \ \text{N}[\text{CASE}=?c, \text{AGR}=?a] \]
\[ \text{VP}[\text{AGR}=?a] \rightarrow \text{IV}[\text{AGR}=?a] \]
\[ \text{VP}[\text{AGR}=?a] \rightarrow \text{TV}[\text{OBJCASE}=?c, \text{AGR}=?a] \ \text{NP}[\text{CASE}=?c] \]

...\[ \text{TV}[\text{OBJCASE}=\text{dat}, \text{AGR}=[\text{NUM}=\text{sg}, \text{PER}=1]] \rightarrow \text{'}folge' \ | \text{'}helfe' \]
\[ \text{TV}[\text{OBJCASE}=\text{dat}, \text{AGR}=[\text{NUM}=\text{sg}, \text{PER}=2]] \rightarrow \text{'}folgst' \ | \text{'}hilfst' \]

...\[ \text{Folgst}/\text{follow} \]
\[ \text{Hilfst}/\text{help} \]
ich folge den Katzen
Features to detect inversions
Many kinds of ‘displaced’ phrases

<table>
<thead>
<tr>
<th>Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Topicalization</td>
<td>This guy, I want XXX to solve the problem</td>
</tr>
<tr>
<td>(2) Wh-question</td>
<td>Who did I want XXX to solve the problem?</td>
</tr>
<tr>
<td>(3) PP-fronting</td>
<td>From which book did the students get the answer XXX?</td>
</tr>
<tr>
<td>(4) Passive</td>
<td>All the ice-cream was eaten XXX</td>
</tr>
<tr>
<td>(5) Right-node raising</td>
<td>I read XXX yesterday a book about China.</td>
</tr>
<tr>
<td>(6) Aux-inversion</td>
<td>Will the guy eat the ice-cream?</td>
</tr>
<tr>
<td>(7) Relative clause</td>
<td>The guy that John saw XXX</td>
</tr>
<tr>
<td>(8) Gapping</td>
<td>I chased XXX and Mary killed XXX a rabid dog.</td>
</tr>
</tbody>
</table>
Questions about displaced phrases

- What kinds of phrases?
- What are configurations for displacement?
- What is ‘distance’ of displacement?
- How can we encode this?
The key configuration: \(c\text{-command}(X,Y)\): the first branching node that dominates \(X\) also dominates \(Y\)

Constraint 1: Fillers must \(c\text{-command}\) their ‘gaps’ (positions from which they are displaced)
I want this student to solve the problem.
Figure 3: Displaced phrases must c-command the canonical positions from which they are displaced. Here, in the sentence 
This student I want to solve the problem,
this student has been displaced from its position as the Subject of to solve, as indicated by the empty element with an index $i$. The displaced NP with index $i$ c-commands the empty element's position.

1. Add new nonterminal rules that map NPs, Ss, PPs, etc. to the empty string, $\varepsilon$. This corresponds to an expansion of a phrase that is then “unpronounced” or silent. For example, for NP we will have the rule, $NP \rightarrow \varepsilon$. Informally, we will call the position of the empty phrase a “gap.” (In fact, this is not exactly the NP expansion rule we want, and we’ll have to modify it slightly as described below.)

2. Add new nonterminal rules corresponding to the introduction of displaced phrase, along with new nonterminal names to keep track of the “link” between the displaced phrase and the place where the empty phrase is. This is done by taking a regular nonterminal, say, $S$ or $VP$, and then creating an additional new non-terminal that has a “slash” in it, with a non-terminal name after the slash that stands for the kind of phrase that has been displaced, and so will eventually expand out as the empty string $\varepsilon$ somewhere below in the expansion of the regular nonterminal. For example, given the “vanilla” rule with nonterminals, $S \rightarrow NP \ VP$, then we can create a “slash” nonterminal: $VP/\ NP$, which denotes the fact that somewhere below in the subtree headed by $VP$, the rule $NP \rightarrow \varepsilon$ must be applied. Of course, it must be the case that there is an NP that gets “paired up” with this unpronounced NP (the “filler” for the “gap). So the new, additional rule for expanding $S$ looks like this: $S \rightarrow NP \ VP/\ NP$. Here, it is tacitly assumed that the first NP (the Subject) is in fact the “filler” for the “gap” that is represented by the slashed nonterminal $VP/\ NP$.

Important: note that the new nonterminal name, $VP/\ NP$ is an atomic symbol, just like $NP$ or $VP$. The purpose of the slash form is to keep track – remember – that an NP has been displaced, and that whatever rules expand out $VP/\ NP$, then somewhere below there must be a matching epsilon-expanding rule.

3. Therefore, whenever a “slash rule” introduces the possibility of a displaced phrase, one must not only add the initial slashed nonterminal and rule, e.g., $VP/\ NP$, one must also add a chain of such rules that terminate in the expansion of NP as $\varepsilon$. For example, given that we have added a new rule $S \rightarrow NP \ VP/\ NP$, one must also add rules that expand $VP/\ NP$ with the slash “passed on” beneath the VP. For example, if we have the vanilla rule expanding $VP$ as, say, $VP NP$, then we must also add a new rule and slash nonterminal, $VP/\ NP \rightarrow VB NP/\ NP$, indicating that the NP “gap” (denoted by the slash) has been passed down to the NP on the right. Finally, we have the rule that terminates the ‘chain’ of slashed rule expansions as $\varepsilon$. For example, in our example sentence, this student I want to solve the problem, where this student has been displaced from its position as the subject of to solve, we...
But never this:

```
CP
  |
  IP
  |
| NP
  |
  \ $\epsilon_i$
| NP
  |
  \ I
    |
    V
      |
      want
        |
        NP
          |
          Det
            |
            N
              |
              this
                |
                student
          |
         to
       |
CP
  |
  IP
  |
| NP
  |
  \ solve
    |
    the
      |
      problem
```
Relative clause

NP

NP\textsubscript{i}

\textsc{Det} \quad \textsc{N} \quad \textsc{Comp} \quad \textsc{IP} \quad \textsc{VP} \quad \textsc{PP} \quad \textsc{NP}

\textit{the} \quad \textit{president} \quad \textit{who} \quad \varepsilon\textit{i} \quad \textit{choked} \quad \textit{on} \quad \textit{the pretzel}
The c-command constraint is central for ‘variable binding’ in human language.

- In the sentence ‘he likes John,’ ‘he’ cannot be ‘John’ because ‘he’ is not dominated by ‘likes’.
- In the sentence ‘he thinks,’ ‘he’ can be ‘John’ because ‘he’ is dominated by ‘thinks’.

This is the concept of coreference, where ‘he’ refers to the same entity in both instances.
How do children learn that a particular interpretation *must* be absent? (How do they pick up on *nothing* in signal?)
Why can ‘he’ be Max in this sentence?
C-command constraint on variable binding resembles environment frames in programming languages.

```
S
  /\  
NP  VP
  |   |
Max said
```

'he' must be free within the box = “Condition C”.

```
S
  /\  
NP  VP
  |   |
he  V  NP
   |   |
ordered sushi
```

‘abstraction barrier’
Constraint on displacement #2: distance

This is OK…what displaced….
Constraint #2: distance

This is NOT OK… *what* displaced how far?
Blocked! Can’t cross two “environment domains”
Can only go to adjacent domain
One more distance example
What about *apparently* long-distance links?
A CFG challenge: can you do this with ‘slash’ categories (1 person has, in 20 yrs)

JOHN GAVE MARY AND BILL TO FRED BOOKS THAT LOOKED REMARKABLY SIMILAR
JOHN HUMMED AND MARY SANG AT EQUAL VOLUMES
JOHN HUMMED AND MARY SANG THE SAME TUNE
I HAVE BEEN WONDERING WHETHER BUT WOULD NOT WANT TO STATE THAT YOUR THEORY IS CORRECT
I CAN TELL YOU WHEN BUT I CAN NOT TELL YOU WHY BILL LEFT ME
I LIKE BUT BILL DOES NOT LIKE TO VISIT NEW CITIES
JOHN HAS CLAIMED BUT I DO NOT BELIEVE THAT BILL IS A COMMUNIST
MARY USED TO BE AND BILL STILL IS VERY SUSPICIOUS
MARY SAID SHE WOULD AND BILL ACTUALLY DID EAT A RAW EGGPLANT
BILL MAY BE AND JOHN CERTAINLY IS A WEREWOLF
BILL OFFERED AND MARY GAVE JOHN A CADILLAC
BILL CAUGHT AND MARY KILLED THE RABID DOG
THE WOMAN WHO WAS HERE BELIEVED THAT THE MAN WAS ILL
THE WOMAN BELIEVED THAT THE GUY WHO WAS HERE WAS ILL
THE WOMAN BELIEVED THAT THE GUY WAS ILL WHO WAS HERE
TO WHICH CITY AND WHICH CONFERENCE DID BILL GO TO
TO WHICH CITY AND TO WHICH CONFERENCE DID BILL GO TO
TO WHICH CITY AND WHICH CONFERENCE DID BILL GO TO TO
TO WHICH CITY AND WHICH CONFERENCE DID BILL GO TO
TO WHICH CITY AND TO WHICH CONFERENCE DID BILL GO
ON WHICH TABLE AND UNDER WHICH FLOWER POT DID MARY PUT THE KEYS
WHERE AND WHEN DID BILL PUT THE BOOK
WHICH BOOK AND WHICH PENCIL DID MARY BUY
MARY ASKED WHO AND WHAT BOUGHT
MARY ASKED WHO AND WHERE BILL HAD SEEN
I WONDER WHO MARY LIKES AND HOPES WILL WIN
I WONDER WHO BILL SAW AND LIKED MARY
I WONDER WHO SAW BILL AND LIKED MARY
I WONDER WHO BILL SAW AND MARY LIKED
Figure 4: An example of how the PTB annotates displaced phrases. Here the sentence is a topicalized form, *The SEC will probably vote on the proposal early next June*, he said. Note the empty phrase under the final S, labeled as a T indexed with a 1. This means it is linked to the S which has been displaced to the front, which also has the index 1, along with a topic annotation, -TPC-.

Require rules that look something like the following, resulting in the parse structure that is displayed in Figure 3. It should be readily apparent that the sequence of new "slash: nonterminals for a chain that runs down from the introduction of the slash nonterminal – always by a rule where it is adjacent to the actual phrase that corresponds to the displaced phrase – all the way down a "spine" until it reaches a point where the phrase is "discharged" by a rule of the form $XP/XP!$. Note: this means that the displaced phrase will always c-command a gap, as it should be in human language.

Question: what happens to the parsing time? (Hint: think about the new grammar size.)

4. It is even possible to incorporate the idea of feature values to work with the "filler and gap" idea. The method is to introduce a new feature, named GAP, which has as its value the name of the phrase that is being displaced. Then it is this GAP feature that must be given a value (by the displaced NP, S, PP, etc.), and we link displaced phrases (and fillers) to their gaps by means of this feature. This turns out to be very tricky to implement; see Figure 5. The reason is that once one posits the possibility of a GAP feature, then it is hard to keep it under control. (It must be able to handle even sentences with displacement to the right, as in *I need, and he wanted, a book about economics*. For another example, in Figure 5, we show the parse tree for *What did Obama eat* and the corresponding parse tree using "slash" notation.

This isn't the only problem with the use of a variable to "pass around" the information about a displaced phrase. Consider example sentences where there are two displaced phrases, as mentioned earlier, where SSS denotes the place from which these sonatas has been displaced, and VVV denotes the place from which which violins is displaced:

*Which violins are these sonatas too difficult to play SSS on VVV*

Meaning: For wh-x, x= violins, and is it too difficult for (someone) to play these sonatas on these violins

Furthermore, examples like these illustrate that there is a third type of "unpronounced" element in some sentences, corresponding to an (arbitrary) person or thing that we can call PRO (for Pronoun, but...
In 1995, Carl de Marcken wrote the first paper on designed a head-centric CFG for use in learning CFGs.

If the reader is interested in a real challenge, the final kind of syntactic structure that poses a di-

Figure 5: The ordinary CFG and "slashed" CFG parse tree for the sentence, "These sonatas promise obama to eat, where violins sonatas for conjunction, but this is not really quite right."

Remarks on Nominalization

The problem is that one can conjoin any two or more phrases of the same type, but representing this is a problem. In the PTB, they used a phrase type introduced by Chomsky, that it is grounded on a Verb. Moreover, the dictionary information is more specific. It is this specific information, as we shall see, that should be used in PCFGs (but is not). This view, introduced by Chomsky from lectures 5-6, nearly 40 years ago, it was determined that phrase name information in human languages is completely redundant with information in the lexicon. Thus, a VP is completely determined by the fact that incorporates what is called lexical "head" information. What is this about? As mentioned in the notes to complete our discussion, and to foreshadow the better use of PCFGs, we introduce the notion of a CFG called 4 CFGs and lexicalized, head-based CFGs.

For the second, augmented CFG system, we must use the discourse. So for example, we have to update the phrase empty position after eat. Thus at some point we must have a new nonterminal that encodes what is referred to earlier in the sentence, if it is in right structural relationship (you should know this by now: the earlier element must c-command the unpronounced).

For almost every approach is conjunctions like CONJ, or even earlier in the first example, and how this empty element is linked...
The Slash 'N Burn Answer

Chain backbone


“Slash” nonterminals encode state
Consequences for parsing?

How does this enforce the c-command constraint?
The Slash `n Burn Answer

CP

Comp

whNP

who

IP/NP

I

did

I'/NP

NP

Obama

VP/NP

V

see

NP/NP

\[\epsilon\]
Feature grammar with slash feature

S[-INV] -> NP VP
S[-INV]/?x -> NP VP/?x
S[-INV] -> NP S/NP
S[-INV] -> Adv[+NEG] S[+INV]
S[+INV] -> V[+AUX] NP VP
S[+INV]/?x -> V[+AUX] NP VP/?x
SBar -> Comp S[-INV]
SBar/?x -> Comp S[-INV]/?x
VP -> V[SUBCAT=intrans, -AUX]
VP -> V[SUBCAT=trans, -AUX] NP
VP/?x -> V[SUBCAT=trans, -AUX]
NP/?x
VP -> V[SUBCAT=clause, -AUX] SBar
VP/?x -> V[SUBCAT=clause, -AUX]
SBar/?x
VP -> V[+AUX] VP
VP/?x -> V[+AUX] VP/?x

V[SUBCAT=intrans, -AUX] -> 'sing'
V[SUBCAT=trans, -AUX] -> 'like'
V[SUBCAT=clause, -AUX] -> 'say'
V[+AUX] -> 'do' | 'can'
NP[-WH] -> 'you' | 'cats'
NP[+WH] -> 'who'
Adv[+NEG] -> 'rarely' | 'never'
NP/NP ->
Comp -> 'that'
Using features for fillers and gaps

“Slash” feature with values NP, PP, etc.

How does this grammar obey c-command?
How does this grammar obey distance constraint?
The Problems with PCFGs

• Two: independence conditions are too strong
• They cannot model non-local dependencies
What it means to be ‘context-free’

- Expansion of a particular rule just depends on itself, not on surrounding context
- Example

Subject | Pronoun | Not Pronoun
--- | --- | ---
Subject | 91% | 0.09%
Object | 35% | 66%

Data from Switchboard (31,021 declarative S’s)
How could we represent this choice?

\[
\begin{align*}
\text{NP} & \rightarrow \text{DT NN} \ 0.28 \\
\text{NP} & \rightarrow \text{PRP} \ 0.25
\end{align*}
\]

But…PCFGs don’t allow conditioning on context….

We want something like this:

\[
\begin{align*}
\text{NP} & \rightarrow \text{DT NN} \ 0.09 \\
\text{NP} & \rightarrow \text{PRP} \ 0.91 \\
\text{NP}_{\text{subject}} & \rightarrow \text{DT NN} \ 0.09 \\
\text{NP}_{\text{object}} & \rightarrow \text{PRP} \ 0.91
\end{align*}
\]
Implementation

S

NP-S

VP-S

he

V-VP

likes

NP-VP

the
guy

“vertical Markovization”

Is this enough? Are there problems? Can we do more?
PCFG Problem #2: not sensitive to actual words

- Wait until last moment to expand words, and this is done independent of context
- But words can help us tell which previous expansions to use…
- Eg: *ate the ice-cream with a fork/ eat the ice-cream with chocolate toppings*
- One solution…
Adding lexical ‘heads’ to CF rules (De Marken, 1995)

\[ S \rightarrow NP(dog) \text{ VP(saw)} \]
But before we can see how to do this..

• We must develop methods for estimating probabilities of productions from pre-parsed corpus

• So far, we have just sketched MLE idea, but it won’t work w/o more clever estimation and smoothing

• First step: see how to do this with linear system, e.g., HMMs.