CSCI 5832
Natural Language Processing

Lecture 11
Jim Martin

Today: 2/22

- More on CFGs and English grammar facts
- Break
- Parsing with CFGs
- Project discussions

Context-Free Grammars

- **Capture constituency and ordering**
  - **Ordering is easy (well not really)**
    What are the rules that govern the ordering of words and bigger units in the language
  - **What's constituency?**
    How words group into units

CFG Examples

- **S -> NP VP**
- **NP -> Det NOMINAL**
- **NOMINAL -> Noun**
- **VP -> Verb**
- **Det -> *a***
- **Noun -> *flight***
- **Verb -> *left***

Problems for CFGs

- Agreement
- Subcategorization
- Movement

Agreement

- This dog
- Those dogs
- *This dogs
- *Those dog

- This dog eats
- Those dogs eat
- *This dog eat
- *Those dogs eats

Agreement

- In English,
  - subjects and verbs have to agree in person and number
  - Determiners and nouns have to agree in number
- Many languages have agreement systems that are far more complex than this.

Possible CFG Solution

- S -> NP VP
- NP -> Det Nominal
- VP -> V NP
- …

- SgS -> SgNP SgVP
- PlS -> PlNp PlVP
- SgNP -> SgDet SgNom
- PlNP -> PlDet PlNom
- PlVP -> PlV NP
- SgVP ->SgV Np
- …

CFG Solution for Agreement

- It works and stays within the power of CFGs
- But its ugly
- And it doesn't scale all that well

Subcategorization

- Sneeze: John sneezed
- Find: Please find [a flight to NY]$_{NP}$
- Give: Give [me]$_{NP}$[a cheaper fare]$_{NP}$
- Help: Can you help [me]$_{NP}$[with a flight]$_{PP}$
- Prefer: I prefer [to leave earlier]$_{TO\text{-}VP}$
- Told: I was told [United has a flight]$_{S}$
- …

Forward Pointer

- It turns out that verb subcategorization facts will provide a key element for semantic analysis (determining who did what to who in an event).

Subcategorization

- *John sneezed the book
- *I prefer United has a flight
- *Give with a flight

- Subcat expresses the constraints that a predicate (verb for now) places on the number and type of the argument it wants to take

So?

- So the various rules for VPs overgenerate.
  - They permit the presence of strings containing verbs and arguments that don't go together
  - For example
  - VP -> V NP therefore

    Sneezed the book is a VP since "sneeze" is a verb and "the book" is a valid NP

Possible CFG Solution

- VP -> V
- VP -> V NP
- VP -> V NP PP
- …

- VP -> IntransV
- VP -> TransV NP
- VP -> TransPP NP PP
- …

Movement

- Core (canonical) example
  - My travel agent booked the flight

Movement

- Core example
  - [[My travel agent]$_{NP}$ [booked [the flight]$_{NP}$]$_{VP}$]$_{S}$
- I.e. "book" is a straightforward transitive verb. It expects a single NP arg within the VP as an argument, and a single NP arg as the subject.

Movement

- What about?
  - Which flight do you want me to have the travel agent book?
- The direct object argument to "book" isn't appearing in the right place. It is in fact a long way from where its supposed to appear.
- And note that its separated from its verb by 2 other verbs.

The Point

- CFGs appear to be just about what we need to account for a lot of basic syntactic structure in English.
- But there are problems
  - That can be dealt with adequately, although not elegantly, by staying within the CFG framework.
- There are simpler, more elegant, solutions that take us out of the CFG framework (beyond its formal power)

Break

Break

- **Quiz**
  - **Average was X**
    1. **True or False**
    2. **Distributional and Morphological Evidence**
    3. **aaa*b* or aa+b* or aa*ab***
    4. **They have a non-zero intersection:**

       The machine in 4 accepts some of strings in L1 but not all.

       It accepts some strings not in L1 as well.

       That is, with respect to L1 it makes two kinds of errors false positives and false negatives.

Break

5 aaaabbb

a)

	a	b
a	3	1
b	0	2

b)

	a	b
a	3	2
b	1	3

Quiz

$c^* = (c + 1) N_{c+1}/N_c$

$c^*_1 = (1 + 1) N_2/N_1$
$= 2 * (1/1)$
$= 2$

Colloquium

- Today's CS colloquium talk is on NLP. That's at 3:30 in ECCR 265.
- There's also an interesting talk tomorrow at noon in Muenzinger E214 on issues related to ML/NLP.

Parsing

- Parsing with CFGs refers to the task of assigning correct trees to input strings
- Correct here means a tree that covers all and only the elements of the input and has an S at the top
- It doesn't actually mean that the system can select the correct tree from among all the possible trees

Parsing

- As with everything of interest, parsing involves a search which involves the making of choices
- We'll start with some basic (meaning bad) methods before moving on to the one or two that you need to know

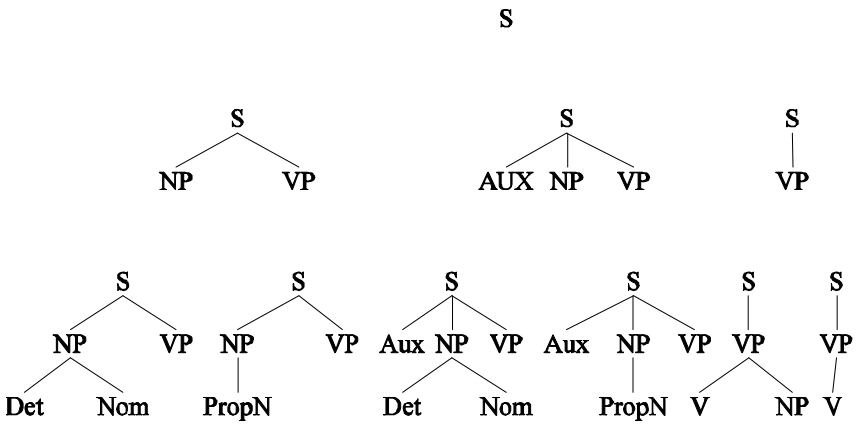
For Now

- Assume…
  - You have all the words already in some buffer
  - The input isn't POS tagged
  - We won't worry about morphological analysis
  - All the words are known

Top-Down Parsing

- Since we're trying to find trees rooted with an S (Sentences) start with the rules that give us an S.
- Then work your way down from there to the words.

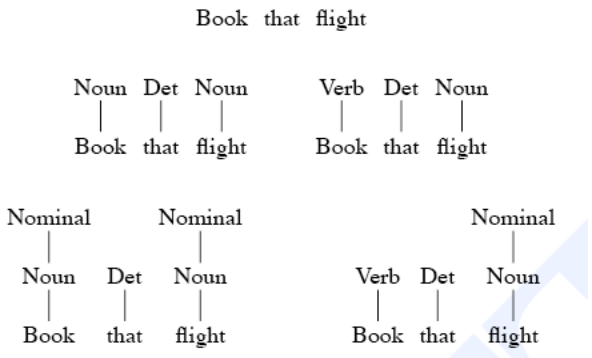
Top Down Space



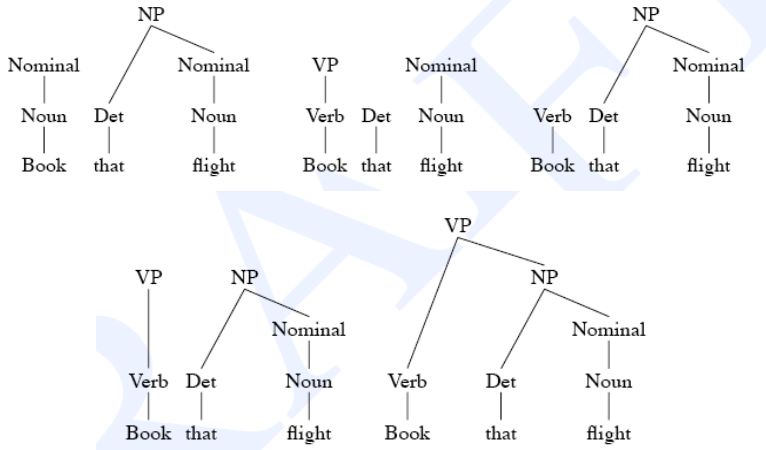
Bottom-Up Parsing

- Of course, we also want trees that cover the input words. So start with trees that link up with the words in the right way.
- Then work your way up from there.

Bottom-Up Space



Bottom Up Space



Control

- **Of course, in both cases we left out how to keep track of the search space and how to make choices**
  - **Which node to try to expand next**
  - **Which grammar rule to use to expand a node**

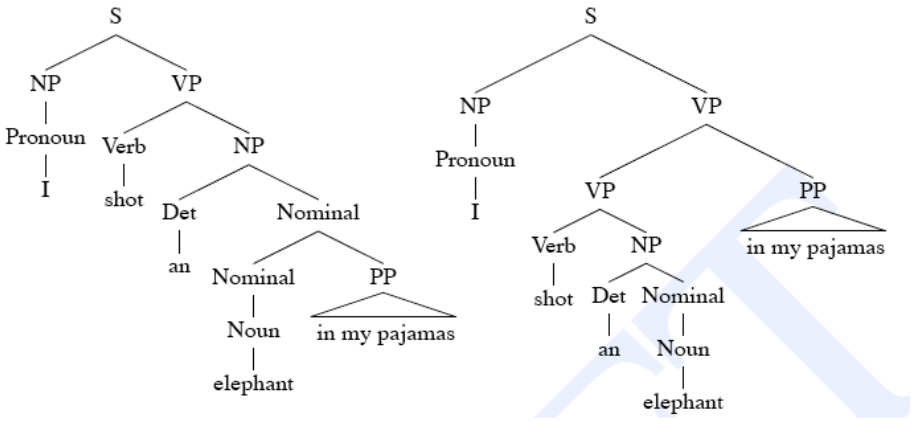
Top-Down and Bottom-Up

- **Top-down**
  - **Only searches for trees that can be answers (i.e. S's)**
  - **But also suggests trees that are not consistent with any of the words**
- **Bottom-up**
  - **Only forms trees consistent with the words**
  - **But suggest trees that make no sense globally**

Problems

- **Even with the best filtering, backtracking methods are doomed if they don't address certain problems**
  - **Ambiguity**
  - **Shared subproblems**

Ambiguity

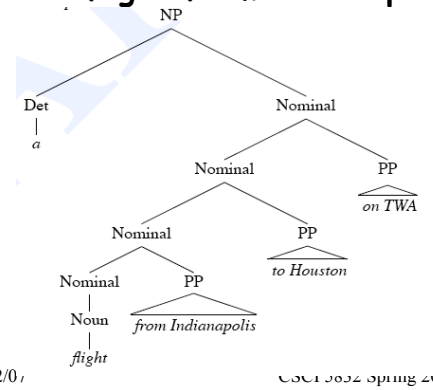


Shared Sub-Problems

- **No matter what kind of search (top-down or bottom-up or mixed) that we choose.**
  - **We don't want to unnecessarily redo work we've already done.**

Shared Sub-Problems

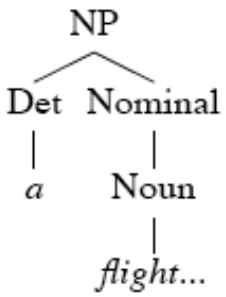
- **Consider**
  - **A flight from Indianapolis to Houston on TWA**



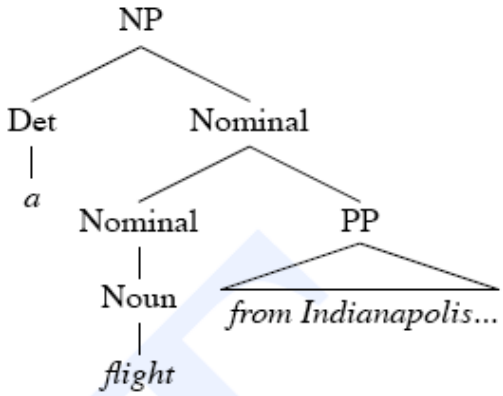
Shared Sub-Problems

- **Assume a top-down parse making bad initial choices on the Nominal rule.**
- **In particular…**
  - **Nominal -> Nominal Noun**
  - **Nominal -> Nominal PP**

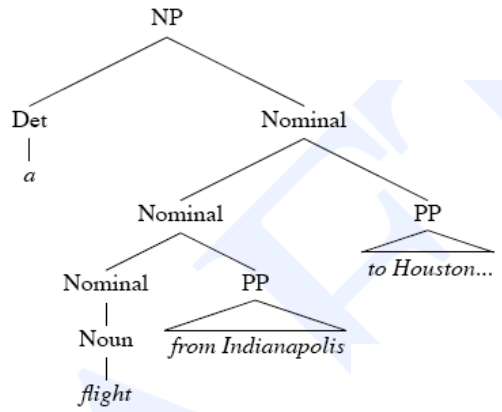
Shared Sub-Problems



Shared Sub-Problems



Shared Sub-Problems



Shared Sub-Problems