Structured predictions: trees and graphs

- Syntactic parsing
  - Phrase structure (or constituent) trees
  - Dependency trees
- Graph-based semantic parsing
  - Abstra
- Needs a gram sible trees for a sentence
- Requires an efficient decoding algorith sequence labeling
- Same statistical or neural models extended to parsing, e.g., Naïve Bayes, Perceptron, Neural networks.

Syntactic Parsing

- Syntactic parsing is the task of assigning a syntactic structure (typically in the form of a tree) to a sentence

- The grammar formalism dictates what a tre l syntactically

- The main tech er NLP proble r a sentence giv

- The general solution to the problem is to bu assigns a "score" to each possible tree, an (search) algorithm to find the best tree (or top-k trees)

- Syntactic parsing is one of the core problems in NLP, and syntactic structures are useful for a wide range of NLP applications

Two broad schools of thought on the syntactic structure

- Phrase structures
  - The building blocks of phrase structures are “phrases” that are consecutive groups of words that make se
  - Each phrase receive a syntactic category (
  - The phrases are hierarchically organize

- Dependenc
  - The fun cture are relations between words in a sentenc consecutive (They are non-consec ot)
  - Dependency relations are thus natur
  - The relations are hierarchically organized
  - The syntactic relations between the words are typically labeled

- Each category of syntactic structures also have different flavors

Phrase-structure parsing based on

Context-free grammars

Context-free languages are specified by context-
which are tuples of (N, Σ, R, S) consistin

- a finite set of
- a finite alphab
- a set of **production rules** R, eac $A \to \beta$, where $A \in N$ and $\beta \in (\Sigma \cup N)^*$
- a designated start symbol S

Example CFG: mathematical operations

$S \rightarrow \mathrm{S\ OP\ S} | \textsc{Num}$

$\mathrm{OP} \rightarrow + | - | \times | \div$

$\textsc{Num} \rightarrow \textsc{Num}\ \textsc{Digit}$

$\textsc{Digit}$ 0 1 2

Is natural language context-free?

- Context-free grammars can be used to represent syntax, which is the set of rules that determine whether an utterance is judged to be grammatical.

- If this representation were perfectly faithful, language such as English could be transform language, co s that would be judg speaker.

- Contemporary theories generally do n languages to be context-free, yet context-free grammars are widely used in natural language parsing.
- The reason is that context-free representations strike a good balance: they cover a broad range of syntactic phenomena, and they can be parsed efficiently.

A phrase-structure grammar

A phrase-structure grammar is one in which sentences are broken down into constituents (phrases), which are contiguous sequences of words that function as coherent units for the purpo linguistic analysis.

Phrases are labeled

heads: noun phrase

S → NP VP
NP → NP PP |we|sushi|chopsticks
NP → NP CC NP
VP → V NP |VP PP
PP → IN NP
V → eat
CC → and

Chomsky Normal Form

- In **Chomsky Normal Form (CNF)** e of every production includes either two non-t terminal symbol.

- CNF can be par

- But some CF NP $\rightarrow$ NP CC NP.

- The general practice in syntactic parsin CNF during training and decoding.

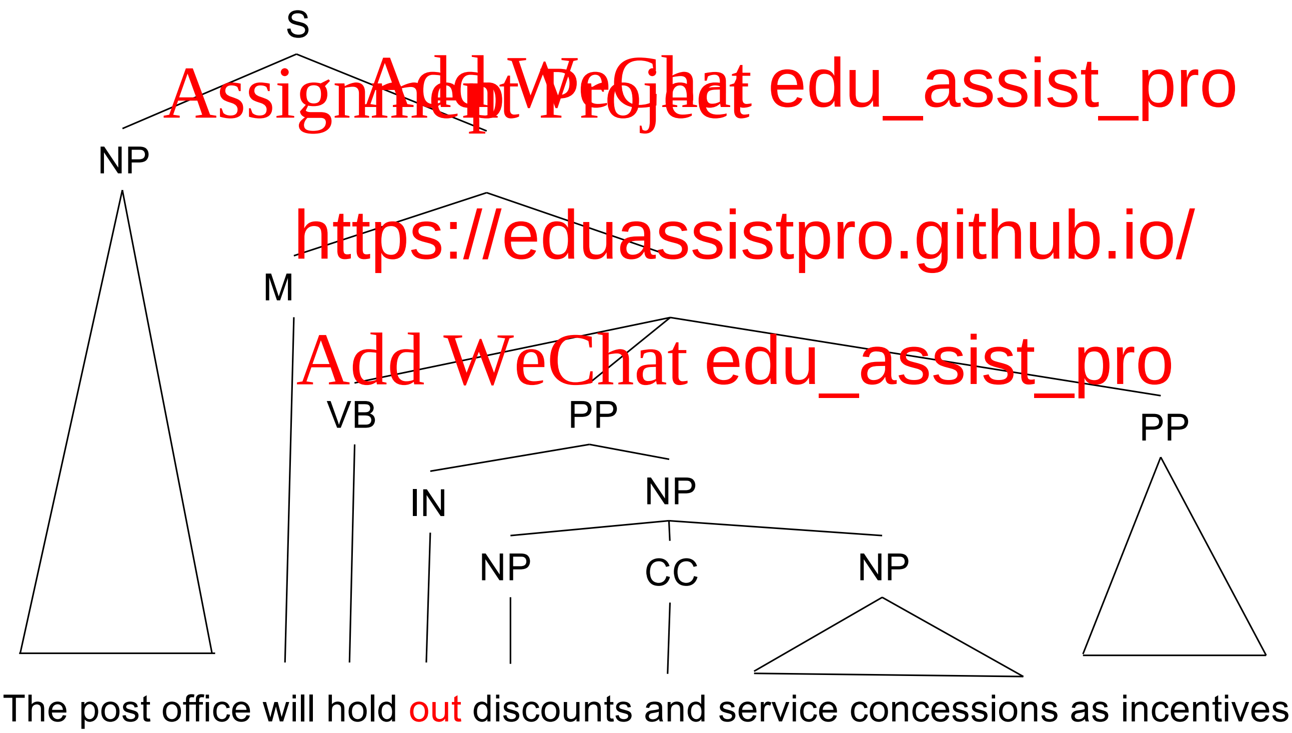
Ambiguity of different types

Even with a grammar, multiple parses (trees) for a sentence are often possible due to ambiguity:

- Attachment ambiguity: e.g., We eat sushi wi… shot an elephant in my pajamas.
- Modifier sco… er
- Particle vs preposition: e.g., The puppy t…
- Complement structure: e.g., The stude… he professor that they didn't understand.
- Coordination scope: e.g., "I see," said the blind man, as he picked up the hammer and saw.

Spurious ambiguity

In practice, a grammar of
with human intuition at all. Such cases are called **spurious ambiguity**



Weighted context-free grammars

The major challenge in sy
possible tree for a senten
possible trees. One way to do that is to have weighted grammar so we can weight the trees...

S → NP VP	S → N	
NP NNP	NP	
NP		-0.5
		-0.5
NP		-0.5
NP → NNS	NP	1
NP → NN	NP	1
NP → NP PP	NP → NP PP	-0.3
VP → VBZ NP	VP → VBZ NP	-0.5
VP → VBD NP	VP → VBD NP	-0.5
VP → VP PP	VP → VP PP	-0.2
PP → IN NP	PP → IN NP	0
ADJP → JJ	ADJP → JJ	0
NNP → John	NNP → John	0

Probabilistic context-free grammars

And one special form of we [illegible]
probabilistic contex[illegible] ll rules
with the same left hand side is 1:

S → NP VP	S → NP VP	1.0
NP → NNP	NP → N	
NP → NP CC NP	NP → N	
[illegible]	NP [illegible] →	
NP [illegible] ADJP NP	NP [illegible] A	
		0.1
		0.1
		0.3
VP → VBZ NP	VP	
VP → VBD NP	VP	
VP → VP PP	VP	
PP → IN NP	PP → IN NP	1.0
ADJP → JJ	ADJP → JJ	1.0
NNP → John	NNP → John	1.0
NNS → apples	NNS → apples	0.4
NNS → oranges	NNS → oranges	0.4
NNS → noodles	NNS → noodles	0.2
NN → gravy	NN → gravy	1.0
JJ → green	JJ → green	1.0
IN → with	IN → with	1.0
VBZ → likes	VBZ → likes	1.0
VBD → ate	VBD → ate	1.0

The probability model of a phrase structure tree

Independence assum

- Place invariance
  probability of the subtree
- Context Free-ness: Probability of a subtree not impacted by words outside of the subtree
- Ancestor Free-ness: Probability of a subtre
  on non-termi

Given these assum
probability of the tree and the input sentence) is:

$$P(\mathcal{T}) = \prod_{i=1}^{m} P(RHS_i | LHS_i)$$

The decoding process is to find the tree that has the highest probability:

$$\hat{\mathcal{T}}(s) = \underset{\mathcal{T} s.t. s=yield(\mathcal{T})}{\text{argmax}} P(\mathcal{T})$$

Probability of an example tree

$P(T_1) = 1.0 \times 0.1 \times 1.0 \times 0.3 \times 1.0 \times 0.3 \times 0.1 \times 0.2 \times 1.0 \times 1.0 \times 0.1 \times 1.0 = 0.000018$

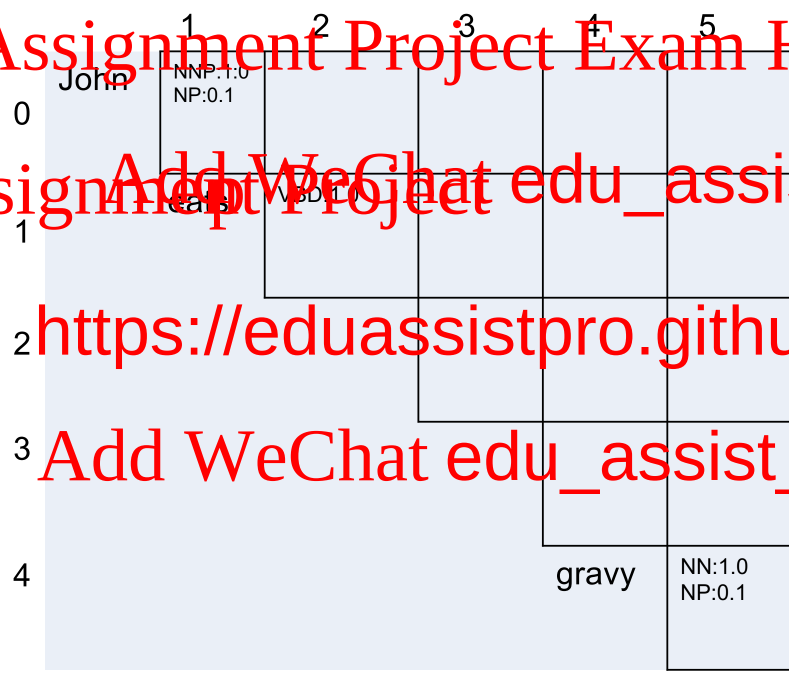
$P(T_2) = 1.0 \times 0.1 \times 1.0 \times 0.5 \times 0.3 \times 1.0 \times 0.1 \times 0.2 \times 1.0 \times 1.0 \times 0.1 \times 1.0 = 0.00003$

Probability of an example tree

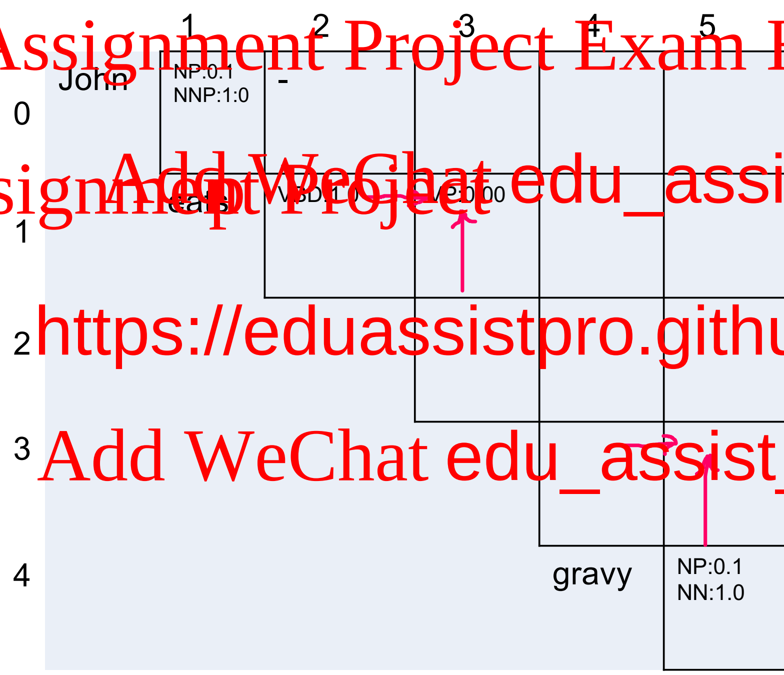
Parsing with PCFGs: the CKY algorithm

procedure W
    for all i, j, X **do**
        $t[i, j, X] \leftarrow 0$
        $b[i, j, X] \leftarrow \emptyset$
    for $m \in \{1, 2, \cdots, M\}$ **do**
        for all
            t ... $m+1, m))$
    for $\ell \in \{2, 3, \cdots, M\}$ **do**
        for $m \in \{0, 1, \cdots, M - \ell\}$
            for $k \in \{m+1, m+2, \cdots, m+\ell-1\}$ **do**
                $t[m, m+\ell, X] \leftarrow \max_{k,Y,Z} \psi(X \rightarrow Y\ Z, (m, m+\ell, k)) + t[m, k, Y] + t[k, m+\ell, Z]$
                $b[m, m+\ell, X] \leftarrow \text{argmax}_{k,Y,Z} \psi(X \rightarrow Y\ Z, (m, m+\ell, k)) + t[m, k, Y] + t[k, m+\ell, Z]$
    return TraceBack(S,0,M,b)

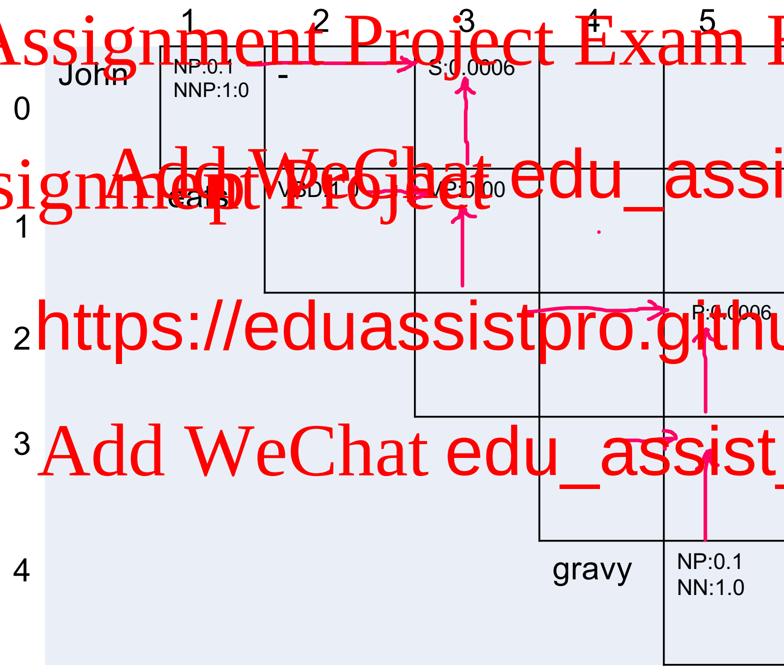
CKY example



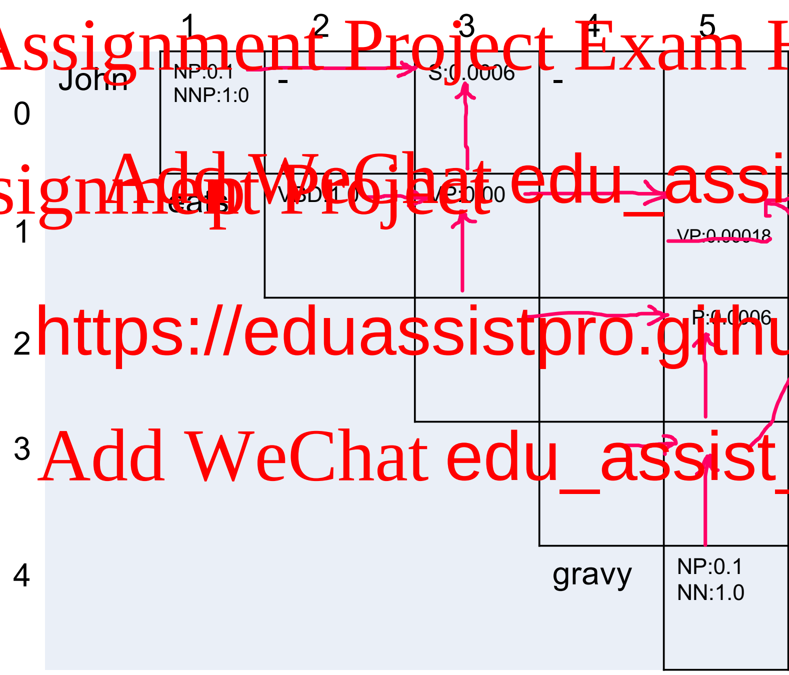
CKY example



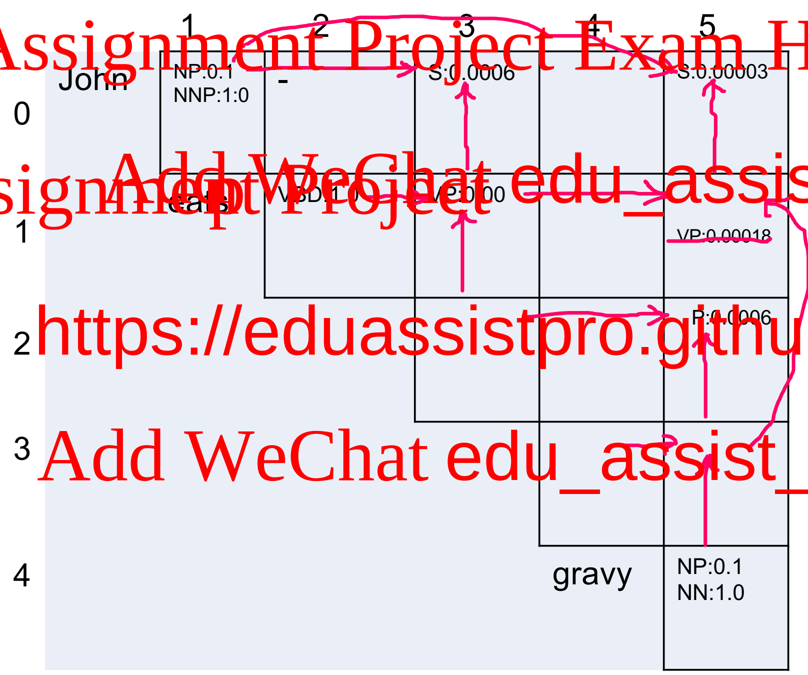
CKY example


1 2 3 4 5
0 1 2 3 4
John
NP:0.1
NNP:1.0
-
S:0.0006
eats
VBD:1.0
gravy
NP:0.1
NN:1.0

CKY example


1
2
3
4
5
0
1
2
3
4
John
NP:0.1
NNP:1.0
-
S:0.0006
-
VBD:1.0
VP:0.00018
gravy
NP:0.1
NN:1.0

CKY example


1
2
3
4
5
John
0
NP:0.1
NNP:1:0
-
S:0.0006
S:0.00003
1
VBD:1.0
VP:0.00018
2
P:0.0006
3
4
gravy
NP:0.1
NN:1.0

Learning PCFGs

Parameters in probabilistic context-free grammars can be estimated by relative frequency, as with HMMs:

$$P(X \to \alpha) = \frac{}{c}$$

E.g., the probability of the production NP $\to$ DET NN is the corpus count of this production, divided by the count of the non-terminal NP. This applies to terminals as well.

Grammar Refinement

- Grammars extracted from treebanks (e.g. TreeBank) are often sensitive to ambiguiti even with the w

- There are vari FG with more expressive productions
  - Parent annotation
  - Lexicalization

Parent annotation

Lexicalized CFGs