

COMS W4705: Natural Language Processing (Fall 2018)

Problem Set #2

Wenbo Gao - wg2313@columbia.edu

October 8, 2018

Problem 1 - PCFGs and HHMs

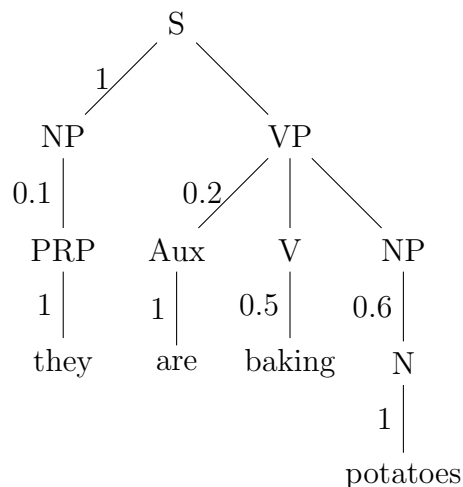
Both PCFGs and HHMs can be seen as generative models that produce a sequence of POS tags and words with some probability (of course the PCFG will generate even more structure, but it will also generate POS tags and words).

(a)

Problem. Revisit the example sentence "*they are baking potatoes*" and grammar from Problem 2. For each sequence of POS tags that is possible for this sentence according to the grammar, what is the joint probability $P(\text{tags}, \text{words})$ according to the PCFG? Hint: consider all parses for the sentence.

Solution.

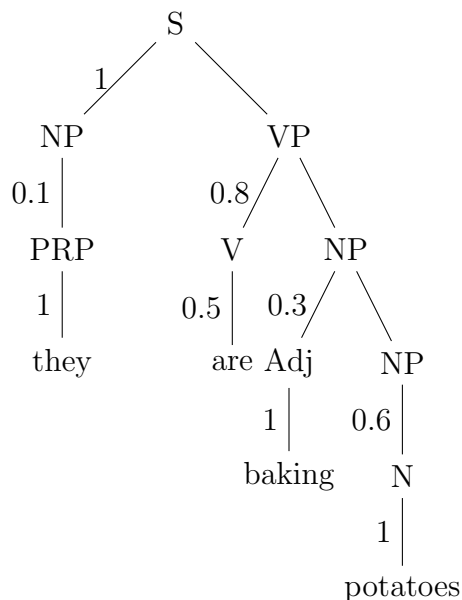
first parse tree:



There is only one possible parse tree (the first parse tree) given tags = {PRP, Aux, V, N}, and words = {they, are, baking, potatoes}, thus

$$\begin{aligned}
 &P[\text{tags} = \{PRP, Aux, V, N\}, \text{words} = \{\text{they}, \text{are}, \text{baking}, \text{potatoes}\}] \\
 &= \sum_{\text{tree} \in \{\text{parse tree with } \{\text{they}, \text{are}, \text{baking}, \text{potatoes}\} \text{ as leaf nodes and } \{PRP, Aux, V, N\} \text{ one layer above}\}} P[\text{tree}] \\
 &= P[\text{first parse tree}] \\
 &= 0.006
 \end{aligned}$$

second parse tree:

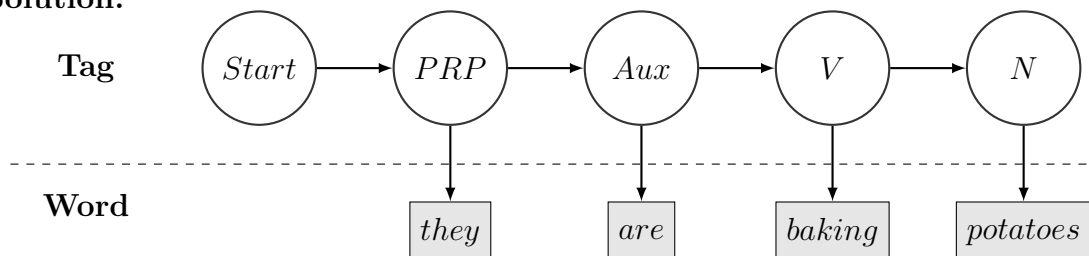


There is only one possible parse tree given tags = { PRP, V, Adj, N}, and words = {they, are, baking, potatoes}, thus

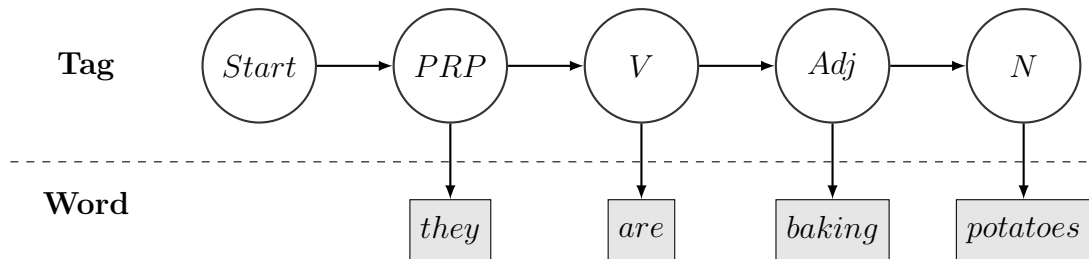
$$\begin{aligned}
 &P[\text{tags} = \{PRP, V, Adj, N\}, \text{words} = \{\text{they}, \text{are}, \text{baking}, \text{potatoes}\}] \\
 &= \sum_{\text{tree} \in \{\text{parse trees with } \{\text{they}, \text{are}, \text{baking}, \text{potatoes}\} \text{ as leaf nodes and } \{PRP, V, Adj, N\} \text{ one layer above}\}} P[\text{tree}] \\
 &= P[\text{second parse tree}] \\
 &= 0.0072
 \end{aligned}$$

(b)

Solution.



$$\begin{aligned}
& P[\text{tags} = \{PRP, Aux, V, N\}, \text{words} = \{they, are, baking, potatoes\}] \\
&= P[\text{tag} = PRP | \text{tag} = \text{Start}] \cdot P[\text{word} = they | \text{tag} = PRP] \\
&\quad \cdot P[\text{tag} = Aux | \text{tag} = PRP] \cdot P[\text{word} = are | \text{tag} = Aux] \\
&\quad \cdot P[\text{tag} = V | \text{tag} = Aux] \cdot P[\text{word} = baking | \text{tag} = V] \\
&\quad \cdot P[\text{tag} = N | \text{tag} = V] \cdot P[\text{word} = potatoes | \text{tag} = N] \\
&= 0.006
\end{aligned}$$



$$\begin{aligned}
& P[\text{tags} = \{PRP, V, Adj, N\}, \text{words} = \{they, are, baking, potatoes\}] \\
&= P[\text{tag} = PRP | \text{tag} = \text{Start}] \cdot P[\text{word} = they | \text{tag} = PRP] \\
&\quad \cdot P[\text{tag} = V | \text{tag} = PRP] \cdot P[\text{word} = are | \text{tag} = V] \\
&\quad \cdot P[\text{tag} = Adj | \text{tag} = V] \cdot P[\text{word} = baking | \text{tag} = Adj] \\
&\quad \cdot P[\text{tag} = N | \text{tag} = Adj] \cdot P[\text{word} = potatoes | \text{tag} = N] \\
&= 0.0072
\end{aligned}$$

Here is my HMM:

1. tag-word transition distributions (the same as in PCFG):

- $P[\text{word} = they | \text{tag} = PRP] = 1$
- $P[\text{word} = are | \text{tag} = Aux] = 1$
- $P[\text{word} = baking | \text{tag} = V] = 0.5, P[\text{word} = are | \text{tag} = V] = 0.5$
- $P[\text{word} = potatoes | \text{tag} = N] = 1$

2. tag-tag transition distributions:

- $P[\text{tag} = PRP | \text{tag} = \text{Start}] = 1$
- $P[\text{tag} = Aux | \text{tag} = PRP] = 0.5, P[\text{tag} = V | \text{tag} = PRP] = 0.5$
- $P[\text{tag} = N | \text{tag} = V] = 0.5, P[\text{tag} = Adj | \text{tag} = V] = 0.5$
- $P[\text{tag} = V | \text{tag} = Aux] = 0.048, P[\text{tag} = PRP | \text{tag} = Aux] = 0.952$
- $P[\text{tag} = N | \text{tag} = Adj] = 0.0576$
- $P[\text{tag} = PRP | \text{tag} = N] = 1$

(c)

Problem. In general, is it possible to translate any PCFG into an HMM that produces the identical *joint* probability $P(\text{tags, words})$ as PCFG (i.e. not just for a single sentence)? Explain how or why not. No formal proof is necessary. Hint: This has nothing to do with probabilities, but with language classes.

Solution. The hidden/latent state space in HMM can be formally described by a finite state machine which is strictly less expressive than pushdown automaton / context-free grammar in PCFG.

Problem 2

Consider the following probabilistic context free grammar.

$S \rightarrow NP VP$	[1.0]
$NP \rightarrow Adj NP$	[0.3]
$NP \rightarrow PRP$	[0.1]
$NP \rightarrow N$	[0.6]
$VP \rightarrow V NP$	[0.8]
$VP \rightarrow Aux V NP$	[0.2]
$PRP \rightarrow they$	[1.0]
$N \rightarrow potatoes$	[1.0]
$Adj \rightarrow baking$	[1.0]
$V \rightarrow baking$	[0.5]
$V \rightarrow are$	[0.5]
$Aux \rightarrow are$	[1.0]

(a)

Problem. Using this grammar, show how the **Earley algorithm** would parse the following sentence.

they are baking potatoes

Write down the complete parse chart. The chart should contain $n + 1$ entries where n is the length of the sentence. Each entry i should contain *all* parser items generated by the parser that end in position i . You can ignore the probability for part (a).

Solution.

Chart[0]

```
> init
s0: S → · NP VP [0,0]
> predict s0, 0|NP|1
s1: NP → · Adj NP [0,0]
s2: NP → · PRP [0,0]
s3: NP → · N [0,0]
> predict s1, 0|Adj|1
s4: Adj → · baking [0,0]
> predict s2, 0|PRP|1
s5: PRP → · they [0,0]
> predict s3, 0|N|1
s6: N → · potatoes [0,0]
> scan s4, 0|baking|1, failure
> scan s5, 0|they|1, success
```

Chart[1]

s7: PRP \rightarrow they \cdot [0,1]
> scan s6, 0|potatoes|₁, failure
> complete with s7, 0|PRP|₁, traverse Chart[0]
. hit s2
s8: NP \rightarrow PRP \cdot [0,1]
> complete with s8, 0|NP|₁, traverse Chart[0]
. hit s0
s9: S \rightarrow NP \cdot VP [0,1]
> predict s9, 1|VP|₂
s10: VP $\rightarrow \cdot$ V NP [1,1]
s11: VP $\rightarrow \cdot$ Aux V NP [1,1]
> predict s10, 1|V|₂
s12: V $\rightarrow \cdot$ baking [1,1]
s13: V $\rightarrow \cdot$ are [1,1]
> predict s11, 1|Aux|₂
s14: Aux $\rightarrow \cdot$ are [1,1]
> scan s12, 1|baking|₂, failure
> scan s13, 1|are|₂, success

Chart[2]

s15: V \rightarrow are \cdot [1,2]
> scan s14, 1|are|₂, success
s16: Aux \rightarrow are \cdot [1,2]
> complete with s15, 1|V|₂, traverse Chart[1]
. hit s10
s17: VP \rightarrow V \cdot NP [1,2]
> complete with s16, 1|Aux|₂, traverse Chart[1]
. hit s11
s18: VP \rightarrow Aux \cdot V NP [1,2]
> predict s17, 2|NP|₃
s19: NP $\rightarrow \cdot$ Adj NP [2,2]
s20: NP $\rightarrow \cdot$ PRP [2,2]
s21: NP $\rightarrow \cdot$ N [2,2]
> predict s18, 2|V|₃
s22: V $\rightarrow \cdot$ baking [2,2]
s23: V $\rightarrow \cdot$ are [2,2]
> predict s19, 2|Adj|₃
s24: Adj $\rightarrow \cdot$ baking [2,2]
> predict s20, 2|PRP|₃
s25: PRP $\rightarrow \cdot$ they [2,2]
> predict s21, 2|N|₃
s26: N $\rightarrow \cdot$ potatoes [2,2]
> scan s22, 2|baking|₃, success

Chart[3]

$s_{27}: V \rightarrow \text{baking} \cdot [2,3]$
 > scan $s_{23}, {}_2|\text{are}|_3$, failure
 > scan $s_{24}, {}_2|\text{baking}|_3$, success
 $s_{28}: \text{Adj} \rightarrow \text{baking} \cdot [2,3]$
 > scan $s_{25}, {}_2|\text{they}|_3$, failure
 > scan $s_{26}, {}_2|\text{potatoes}|_3$, failure
 > complete with $s_{27}, {}_2|V|_3$, traverse Chart[2]
 . hit s_{18}
 $s_{29}: \text{VP} \rightarrow \text{Aux } V \cdot \text{NP} [1,3]$
 > complete with $s_{28}, {}_2|\text{Adj}|_3$, traverse Chart[3]
 . hit s_{19}
 $s_{30}: \text{NP} \rightarrow \text{Adj} \cdot \text{NP} [2,3]$
 > predict $s_{29}, {}_3|\text{NP}|_4$
 $s_{31}: \text{NP} \rightarrow \cdot \text{Adj NP} [3,3]$
 $s_{32}: \text{NP} \rightarrow \cdot \text{PRP} [3,3]$
 $s_{33}: \text{NP} \rightarrow \cdot \text{N} [3,3]$
 > predict $s_{30}, {}_3|\text{NP}|_4$, duplicate
 > predict $s_{31}, {}_3|\text{Adj}|_4$
 $s_{34}: \text{Adj} \rightarrow \cdot \text{baking} [3,3]$
 > predict $s_{32}, {}_3|\text{PRP}|_4$
 $s_{35}: \text{PRP} \rightarrow \cdot \text{they} [3,3]$
 > predict $s_{33}, {}_3|\text{N}|_4$
 $s_{36}: \text{N} \rightarrow \cdot \text{potatoes} [3,3]$
 > scan $s_{34}, {}_3|\text{baking}|_4$, failure
 > scan $s_{35}, {}_3|\text{they}|_4$, failure
 > scan $s_{36}, {}_3|\text{potatoes}|_4$, success

Chart[4]

$s_{37}: \text{N} \rightarrow \text{potatoes} \cdot [3,4]$
 > complete with $s_{37}, {}_3|\text{N}|_4$, traverse Chart[3]
 . hit s_{33}
 $s_{38}: \text{NP} \rightarrow \text{N} \cdot [3,4]$
 > complete with $s_{38}, {}_3|\text{NP}|_4$, traverse Chart[3]
 . hit s_{29}
 $s_{39}: \text{VP} \rightarrow \text{Aux } V \text{ NP} \cdot [1,4]$
 . hit s_{30}
 $s_{40}: \text{NP} \rightarrow \text{Adj NP} \cdot [2,4]$
 > complete with $s_{39}, {}_1|\text{VP}|_4$, traverse Chart[1]
 . hit s_9
 $s_{41}: \text{S} \rightarrow \text{NP VP} \cdot [0,4]$, **solution 1**(s_9, s_{39})
 > complete with $s_{40}, {}_2|\text{NP}|_4$, traverse Chart[2]
 . hit s_{17}
 $s_{42}: \text{VP} \rightarrow \text{V NP} \cdot [1,4]$
 > complete with $s_{42}, {}_1|\text{VP}|_4$, traverse Chart[1]

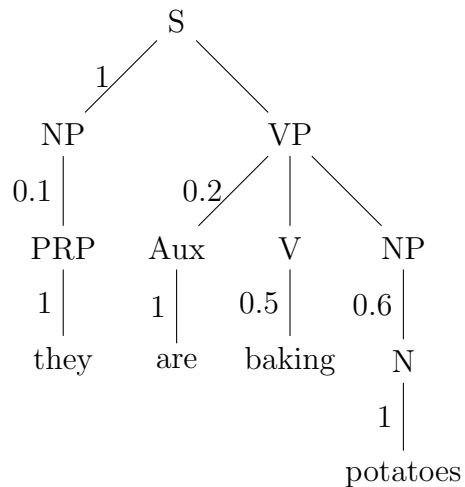
. hit s_9

$S \rightarrow NP VP \cdot [0,4]$, duplicate of s_{41} , **solution 2**(s_9, s_{42})

(b)

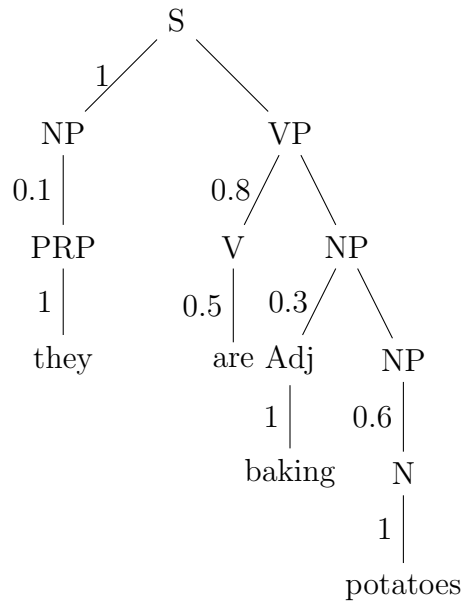
Problem. Write down *all* parse trees for the sentence and grammar from problem 2 and compute their probabilities according to the PCFG.

Solution. first parse tree:



$$\begin{aligned}
 &P[\text{first parse tree}] \\
 &= P[S \rightarrow NP VP] \cdot P[NP \rightarrow PRP] \cdot P[VP \rightarrow Aux V NP] \cdot P[PRP \rightarrow they] \\
 &\quad \cdot P[Aux \rightarrow are] \cdot P[V \rightarrow baking] \cdot P[NP \rightarrow N] \cdot P[N \rightarrow potatoes] \\
 &= 1 \cdot 0.1 \cdot 0.2 \cdot 1 \cdot 1 \cdot 0.5 \cdot 0.6 \cdot 1 \\
 &= 0.006
 \end{aligned}$$

second parse tree:



$$\begin{aligned}
 &P[\text{second parse tree}] \\
 &= P[S \rightarrow NP VP] \cdot P[NP \rightarrow PRP] \cdot P[VP \rightarrow V NP] \cdot P[PRP \rightarrow they] \\
 &\quad \cdot P[V \rightarrow are] \cdot P[NP \rightarrow Adj NP] \cdot P[Adj \rightarrow baking] \cdot P[NP \rightarrow N] \cdot P[N \rightarrow potatoes] \\
 &= 1 \cdot 0.1 \cdot 0.8 \cdot 1 \cdot 0.5 \cdot 0.3 \cdot 1 \cdot 0.6 \cdot 1 \\
 &= 0.0072
 \end{aligned}$$

Problem 3 - CKY parsing

(a)

Problem. Convert the grammar from problem 2 into an equivalent grammar in Chomsky Normal Form (CNF). Write down the new grammar. Also explain what the general rule is for dealing with

1. Rules of the form $A \rightarrow B$ (i.e. a single non-terminal on the right hand side).
2. Rules with three or more non-terminals on the right hand side (e.g. $A \rightarrow B C D E$).

You don't have to deal with the case in which terminals and non-terminals are mixed in a rule right-hand side. You also do not have to convert the probabilities. Hint: Think about adding new non-terminal symbols.

Solution.

New grammar:

$S \rightarrow NP VP$	[0.3]
$S \rightarrow PRP VP$	[0.1]
$S \rightarrow N VP$	[0.6]
$NP \rightarrow Adj NP$	[0.3]
$NP \rightarrow Adj PRP$	[0.1]
$NP \rightarrow Adj N$	[0.6]
$VP \rightarrow V NP$	[0.24]
$VP \rightarrow V PRP$	[0.08]
$VP \rightarrow V N$	[0.48]
$VP \rightarrow AV NP$	[0.06]
$VP \rightarrow AV PRP$	[0.02]
$VP \rightarrow AV N$	[0.12]
$AV \rightarrow Aux V$	[1.0]

General rules:

1. for $A \rightarrow B$
 - find all rules with A on the right-hand side, e.g. $C \rightarrow A D$
 - for each of them, substitute A on the right-hand side by B , i.e. $C \rightarrow B D$
 - add the modified rules to the rule set, and remove the original rule
2. for $A \rightarrow B C D E$
 - add non-terminals $\{ F, G \}$
 - add rules:
 - (a) $A \rightarrow B F$

(b) $F \rightarrow C G$

(c) $G \rightarrow D E$

- remove the original rule

(b)

Problem. Using your grammar, fill the CKY parse chart as shown in class and show all parse trees.

Solution. See Figure 1.

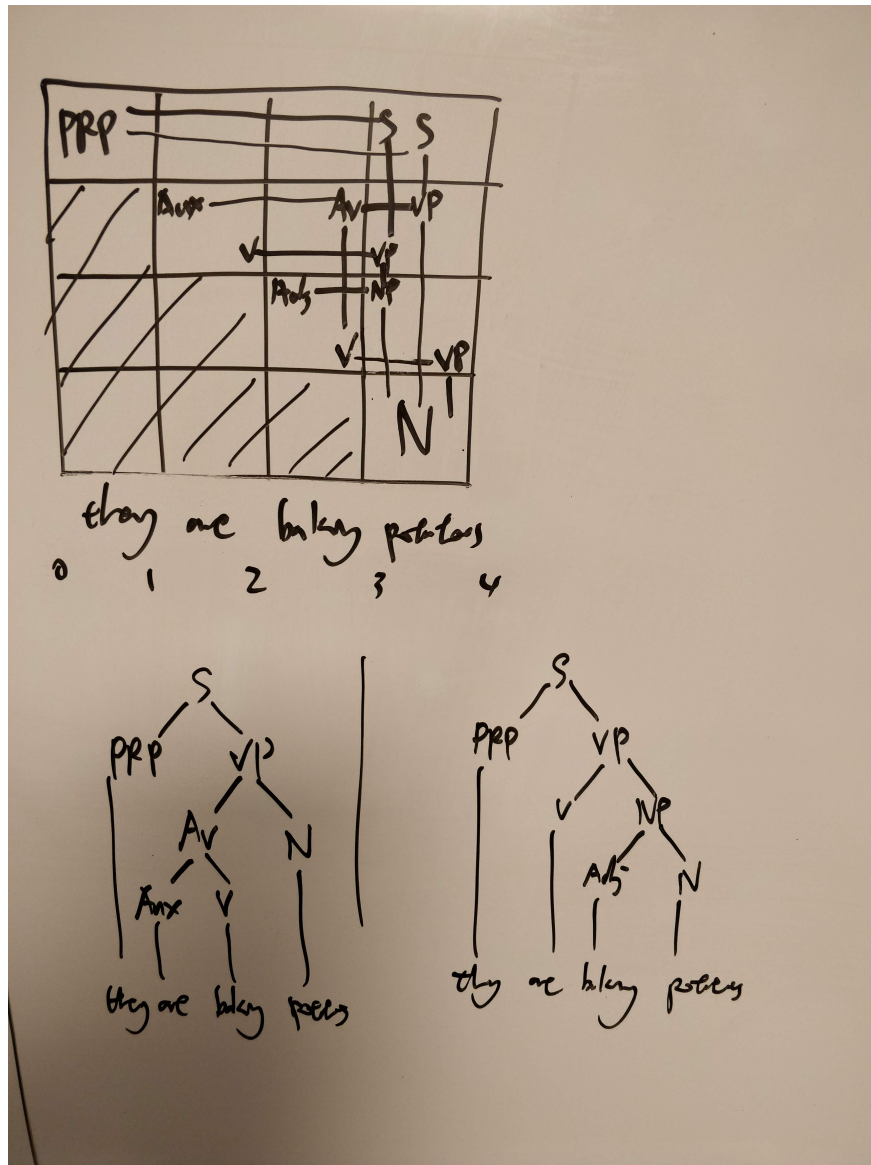


Figure 1: problem 3 (b)