Natural Language Processing – 097215 - Homework Assignment - Dry 2

Your answers will be evaluated thoroughly. Prove everything you claim, and leave no space for mistakes.

*** Notice, there are 6 questions in this assignment, you need to submit only 5 of them (you can choose between q5 and q6).**

Question 1

Say we have a PCFG with start symbol S, and the following rules with associated probabilities:

- q(S $\rightarrow$ NP VP) = 1.0
- q(VP $\rightarrow$ Vt NP) = 1.0
- q(Vt $\rightarrow$ saw) = 1.0
- q(NP $\rightarrow$ John) = 0.25
- q(NP $\rightarrow$ DT NN) = 0.25
- q(NP $\rightarrow$ NP CC NP) = 0.3
- q(NP $\rightarrow$ NP PP) = 0.2
- q(DT $\rightarrow$ the) = 1.0
- q(NN $\rightarrow$ dog) = 0.25
- q(NN $\rightarrow$ cat) = 0.25
- q(NN $\rightarrow$ house) = 0.25
- q(NN $\rightarrow$ mouse) = 0.25
- q(CC $\rightarrow$ and) = 1.0
- q(PP $\rightarrow$ IN NP) = 1.0
- q(IN $\rightarrow$ with) = 0.5
- q(IN $\rightarrow$ in) = 0.5

Denote S_1 = "John saw the cat and the dog with the mouse".

Claim:

"All parse trees for S_1 have either probability C under the given PCFG (for some $C > 0$)"

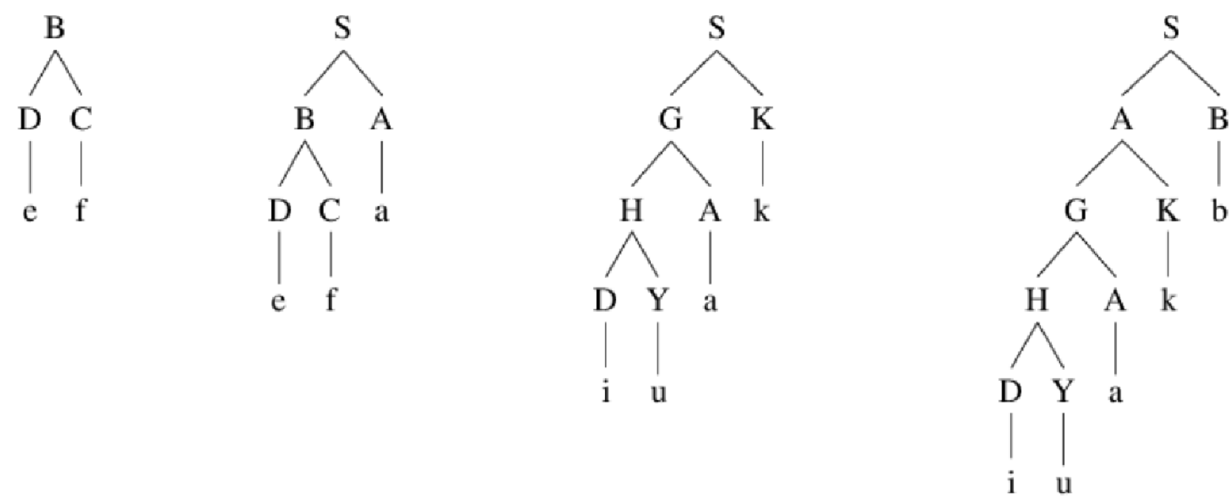
Prove the claim and find C.

Question 2

Consider the CKY algorithm for parsing with PCFGs. The usual recursive definition in this algorithm is as follows:

$$\pi(i,j,X) = \max_{\substack{X\to Y\ Z\in R,\\ s\in\{i\ldots(j-1)\}}} \left(q(X\to Y\ Z)\times \pi(i,s,Y)\times\pi(s+1,j,Z)\right)$$

Now we would like to modify the CKY parsing algorithm to that it returns the maximum probability for any "left-branching" tree for an input sentence. Here are some example left-branching trees:

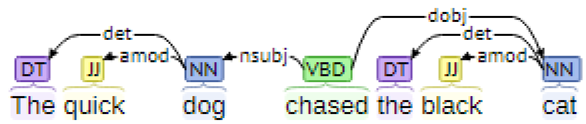


It can be seen that in left-branching trees, whenever a rule of the form X-> Y Z is seen in the tree, then the non-terminal Z must directly dominate a terminal symbol.
Assuming that our goal is to find the highest probability left-branching tree.

Write all recursive definitions and explain it (make sure you cover all necessary details).

Question 3

Given the following sentence and a parse tree, show the states of the Transition-Based (MALT) Parser:



Question 4

In tutorial 8, we saw the Chu-Liu-Edmonds algorithm, employed as an inference algorithm in a graph-based dependency parser.

For a given sentence $x = x_1, x_2, ..., x_n$:

- The directed graph $G_x = (V_x, E_x)$ is given by:

  $V_x = \{x_0 = root, x_1, x_2, ..., x_n\}$

  $E_x = \{(i, j): i \neq j, (i, j) \in [0{:}n] \times [1{:}n]\}$

Prove:

a. The algorithm terminates after a finite number of steps.
b. The algorithm's overall time complexity is $O(V_x \cdot E_x)$, given that:
   - $|V_x| \leq |E_x|$.
   - The complexity for finding a cycle is $O(E_x + V_x)$.

Choose only one of the following questions (5 or 6) and submit.

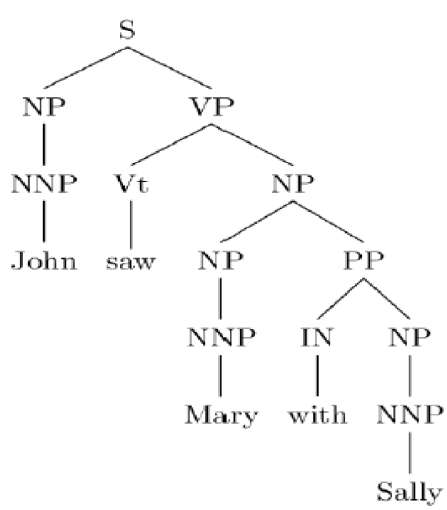
Question 5

Say we have a PCFG with the following rules and probabilities:

- $q(\text{S} \rightarrow \text{NP VP}) = 1.0$
- $q(\text{VP} \rightarrow \text{Vt NP}) = 0.2$
- $q(\text{VP} \rightarrow \text{VP PP}) = 0.8$
- $q(\text{NP} \rightarrow \text{NNP}) = 0.8$
- $q(\text{NP} \rightarrow \text{NP PP}) = 0.2$
- $q(\text{NNP} \rightarrow \text{John}) = 0.2$
- $q(\text{NNP} \rightarrow \text{Mary}) = 0.3$
- $q(\text{NNP} \rightarrow \text{Sally}) = 0.5$
- $q(\text{PP} \rightarrow \text{IN NP}) = 1.0$
- $q(\text{IN} \rightarrow \text{with}) = 1.0$
- $q(\text{Vt} \rightarrow \text{saw}) = 1.0$

Consider the sentence: “John saw Mary with Sally”.

The gold-standard parse tree for this sentence is:



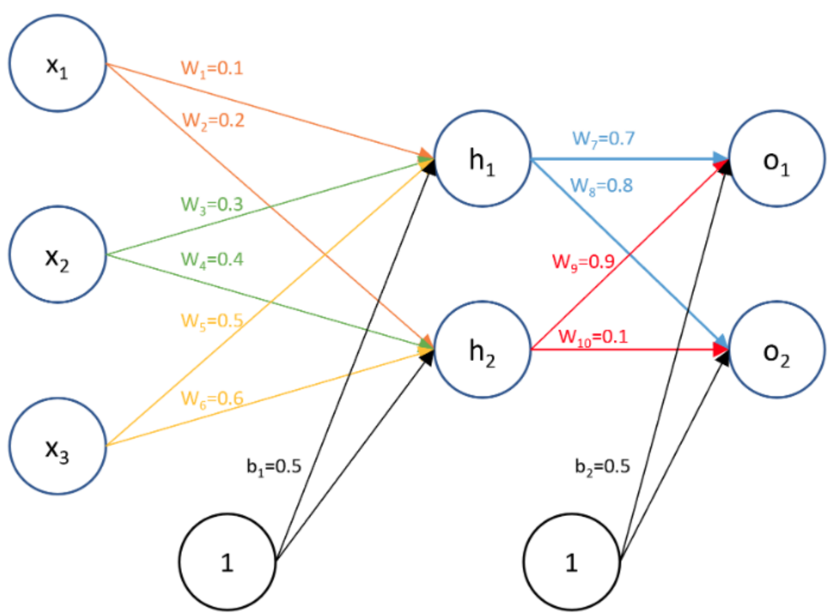
Run the CKY algorithm for the sentence “John saw Marry with Sally” in detail and produce the resulting parse tree under the given grammar.

Compute the F-score for the resulting parse tree.

(Note: the grammar is not in Chomsky normal form, so you must add a few improvements to the CKY algorithm. Describe this improvements in detail.)

Question 6

We will now solve a simple backpropagation exercise. Consider the following network:



Notice the value of each weight is given within the figure (plus bias weights).
Let's assume that our dataset consist of one sample, with the following values:

- Input: $x = [1, 4, 5]^T \in \mathbb{R}^3$
- Gold Output: $y = [0.1, 0.05]^T \in \mathbb{R}^2$
- Number of hidden layers: 1
- Activation: sigmoid (for both hidden and output layers)
- Loss function: MSE

1) Calculate the forward pass of the network with the specified example. What is the error of the model?
2) Calculate the backward pass of the model with respect to the following weights:
   - w_1
   - w_9

- Reminders:
  - Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$
  - Sigmoid derivative: $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$