1.5em 0pt

COMS W4705–Spring, 2021
Name: Weiyao Xie
UNI: wx2251

HW2

1. **Solution to Problem 2.1**

   I tried two extensions in 1.2.3.

   The first extensions I tried is changing the RNN layer to a bi-LSTM layer. There are several advantages that LSTM has over RNN. LSTM is good at long term memory. Although, theoretically RNN can process long distance words, in practice, RNN struggle to achieve good performance when the sentence is long. This will hurt our f1 score because we have a lot long sentences in our twitter data. LSTM introduces different gates to solve this problem. Each gate has two activation functions, sigmoid and tanh. Sigmoid is in charge of making decisions and tanh is in charge of updating the values. For example, in the input gate, sigmoid function output 1 for parameters we want to update, 0 for which we do not want to update. Tanh then produces the values to update the parameters. The input gate will help the model to decide which parts of the previous layer is still important in the future training, and the forget gate let the model forget the part that is less important. Another advantage is that bi-LSTM can look go through the sentences from beginning to end, and also from end to beginning. With this extra context information, the model performance can be improved.

   The second extensions I tried is some extra data preprocessing. First, I convert all urls into ||URL||. I think we cannot extract a lot of useful information from URLs here, but instead of simply converting all urls into <UNK> tag, using a unique tag for all URLs may be beneficial because sometimes posting URLs can signify someone's emotions and we are not loosing all information. Similar reason for user names. I convert user names starting with @ into ||USER|| to preserve some useful information. Besides these preprocessing changes, I used the same architecture from extension1. The performance is slightly better than extension1.

2. **Solution to Problem 2.2**

2.2.1 Forward Propagation

a.

To get h_1 we need q_1:

$$\begin{aligned} q_1 &= W_x^T x_1 + W_h^T h_0 + b_h \\ &= \begin{bmatrix} 1 \\ \\ 3 \end{bmatrix} * -1 + \begin{bmatrix} 1 & 2 \\ \\ 3 & 1 \end{bmatrix} * \begin{bmatrix} 2 \\ \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ \\ 2 \end{bmatrix} \\ &= \begin{bmatrix} 4 \\ \\ 6 \end{bmatrix} \end{aligned} \tag{1}$$

$$\begin{aligned} h_1 &= f(q_1) \\ &= \begin{bmatrix} \frac{1}{1+\exp -4} \\ \\ \frac{1}{1+\exp -6} \end{bmatrix} \end{aligned} \tag{2}$$

b.

To get $\hat{y}_1$ we need p_1:

$$\begin{aligned} p_1 &= W_y^T h_1 + b_y \\ &= \begin{bmatrix} 3 & 1 \end{bmatrix} * \begin{bmatrix} \frac{1}{1+\exp -4} \\ \\ \frac{1}{1+\exp -6} \end{bmatrix} + 1 \\ &= \frac{3}{1+\exp -4} + \frac{1}{1+\exp -6} + 1 \end{aligned} \tag{3}$$

$$\hat{y}_1 = f(p_1) = 0.993 \tag{4}$$

c.

To get h_2, we need q_2:

$$
\begin{aligned}
q_2 &= W_x^T x_2 + W_h^T h_1 + b_h \\
&= \begin{bmatrix} 1 \\ \\ 3 \end{bmatrix} * 0 + \begin{bmatrix} 1 & 2 \\ \\ 3 & 1 \end{bmatrix} * \begin{bmatrix} \frac{1}{1+\exp -4} \\ \\ \frac{1}{1+\exp -6} \end{bmatrix} + \begin{bmatrix} 1 \\ \\ 2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{1+\exp -4} + \frac{2}{1+\exp -6} + 1 \\ \\ \frac{3}{1+\exp -4} + \frac{1}{1+\exp -6} + 2 \end{bmatrix}
\end{aligned} \tag{5}
$$

$$
\begin{aligned}
h_2 &= f(q_2) \\
&= \begin{bmatrix} 0.982 \\ \\ 0.998 \end{bmatrix}
\end{aligned} \tag{6}
$$

d.

To get $\hat{y}_1$ we need p_1:

$$
\begin{aligned}
p_2 &= W_y^T h_2 + b_y \\
&= \begin{bmatrix} \\ 3 \quad 1 \end{bmatrix} * \begin{bmatrix} 0.98 \\ \\ 0.99 \end{bmatrix} + 1 \\
&= 4.94
\end{aligned} \tag{7}
$$

$$
\hat{y}_2 = f(p_2) = 0.993 \tag{8}
$$

2.

$$
\begin{aligned}
L &= \frac{1}{2}((0 - 0.99)^2 + (1 - 0.99)^2) \\
&= 0.493
\end{aligned} \tag{9}
$$

2.2.2 Backpropagation Through Time

1. RNN Computation Graph

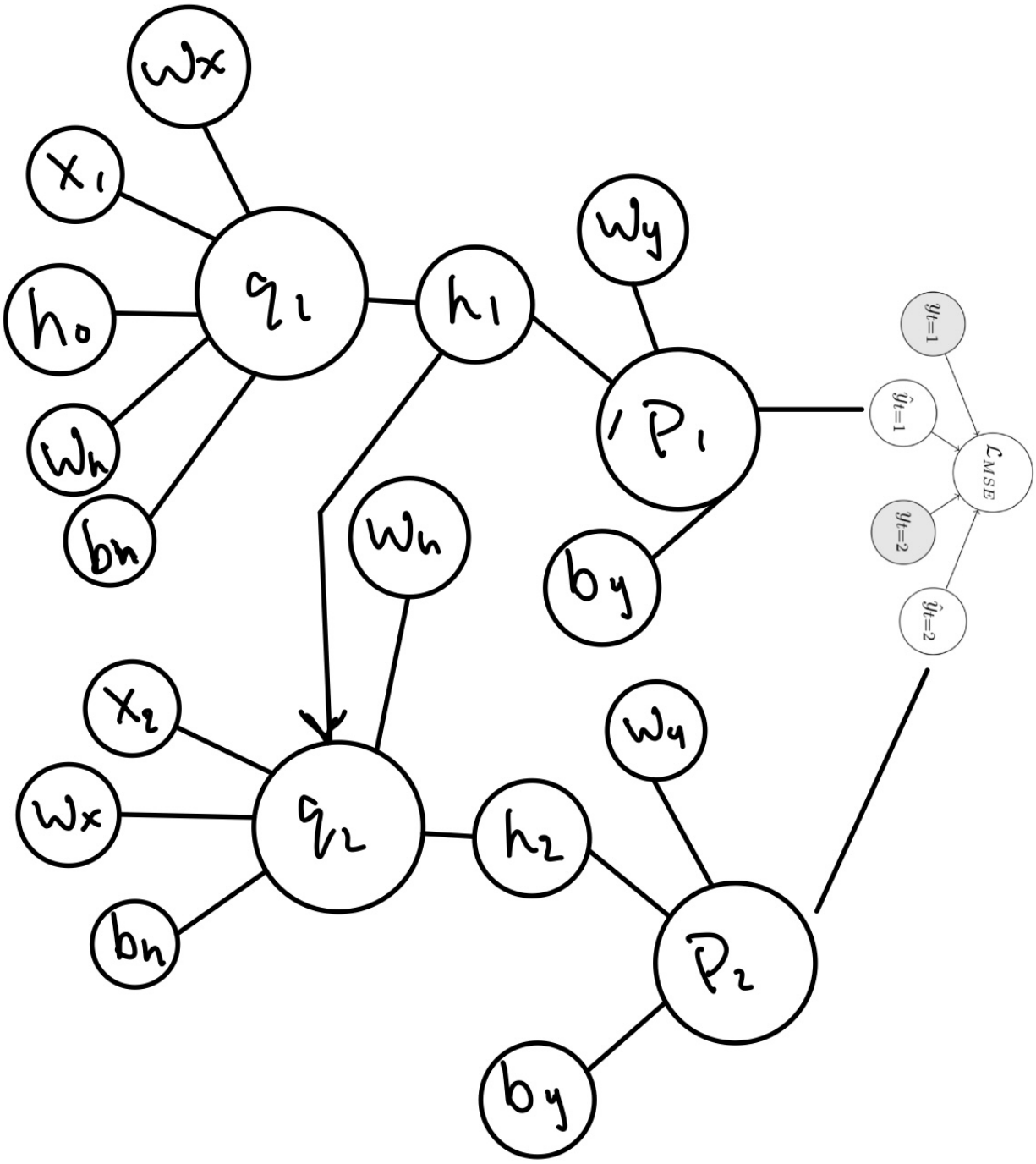


Figure 1: RNN Computation Graph

2. Backpropagation

a.

$$\begin{aligned}\frac{\partial L}{\partial W_y} &= \sum \frac{\partial L}{\partial y} * \frac{\partial y}{\partial p} * \frac{\partial p}{\partial W_y} \\ &= \frac{\partial L}{\partial y_1} * \frac{\partial y_1}{\partial p_1} * \frac{\partial p_1}{\partial W_y} + \frac{\partial L}{\partial y_1} * \frac{\partial y_1}{\partial p_1} * \frac{\partial p_1}{\partial W_y} \\ &= (\hat{y_1} - y_1)[\sigma(p_1)(1-\sigma(p_1))] * h_1 + (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))] * h_2 \\ &= \begin{bmatrix} 0.0068 \\ \\ 0.0069 \end{bmatrix}\end{aligned} \tag{10}$$

b.

$$\begin{aligned}\frac{\partial L}{\partial W_h} &= \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial p_1}\frac{\partial p_1}{\partial h_1}\frac{\partial h_1}{\partial q_1}\frac{\partial q_1}{\partial W_h} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial h_2}\frac{\partial h_2}{\partial q_2}\frac{\partial q_2}{\partial W_h} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial h_2}\frac{\partial h_2}{\partial q_2}\frac{\partial q_2}{\partial h_1}\frac{\partial h_1}{\partial q_1}\frac{\partial q_1}{\partial W_h} \\ &= (\hat{y_1} - y_1)[\sigma(p_1)(1-\sigma(p_1))]W_y^T[\sigma(q_1)(1-\sigma(q_1))]h_1^T \\ &+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))]W_y^T[\sigma(q_2)(1-\sigma(q_2))]h_2^T \\ &+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))]W_y^T[\sigma(q_2)(1-\sigma(q_2))] * W_h^T * [\sigma(q_1)(1-\sigma(q_1))] * h_1^T \\ &= \begin{bmatrix} 7.37 * 10^{-4} & 3.43 * 10^{-5} \\ \\ 3.67 * 10^{-4} & 1.71 * 10^{-5} \end{bmatrix}\end{aligned} \tag{11}$$

c.

$$
\begin{aligned}
\frac{\partial L}{\partial W_x} &= \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial p_1}\frac{\partial p_1}{\partial h_1}\frac{\partial h_1}{\partial q_1}\frac{\partial q_1}{\partial W_x} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial h_2}\frac{\partial h_2}{\partial q_2}\frac{\partial q_2}{\partial W_x} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial h_2}\frac{\partial h_2}{\partial q_2}\frac{\partial q_2}{\partial h_1}\frac{\partial h_1}{\partial q_1}\frac{\partial q_1}{\partial W_x} \\
&= (\hat{y_1} - y_1)[\sigma(p_1)(1-\sigma(p_1))]W_y^T[\sigma(q_1)(1-\sigma(q_1))]x_1 \\
&+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))]W_y^T[\sigma(q_2)(1-\sigma(q_2))]x_2 \\
&+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))]W_y^T[\sigma(q_2)(1-\sigma(q_2))] * W_h^T * [\sigma(q_1)(1-\sigma(q_1))] * x_1 \\
&= \begin{bmatrix} -3.70 * 10^{-4} & -1.72 * 10^{-5} \end{bmatrix}
\end{aligned} \tag{12}
$$

d.

$$
\begin{aligned}
\frac{\partial L}{\partial b_y} &= \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial p_1}\frac{\partial p_1}{\partial b_y} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial b_y} \\
&= (\hat{y_1} - y_1)[\sigma(p_1)(1-\sigma(p_1))] * 1 \\
&+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))] * 1 \\
&= 0.0069
\end{aligned} \tag{13}
$$

e.

$$
\begin{aligned}
\frac{\partial L}{\partial b_h} &= \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial p_1}\frac{\partial p_1}{\partial h_1}\frac{\partial h_1}{\partial q_1}\frac{\partial q_1}{\partial b_h} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial h_2}\frac{\partial h_2}{\partial q_2}\frac{\partial q_2}{\partial b_h} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial p_2}\frac{\partial p_2}{\partial h_2}\frac{\partial h_2}{\partial q_2}\frac{\partial q_2}{\partial h_1}\frac{\partial h_1}{\partial q_1}\frac{\partial q_1}{\partial b_h} \\
&= (\hat{y_1} - y_1)[\sigma(p_1)(1-\sigma(p_1))]W_y^T[\sigma(q_1)(1-\sigma(q_1))] * 1 \\
&+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))]W_y^T[\sigma(q_2)(1-\sigma(q_2))] * 1 \\
&+ (\hat{y_2} - y_2)[\sigma(p_2)(1-\sigma(p_2))]W_y^T[\sigma(q_2)(1-\sigma(q_2))]W_h^T[\sigma(q_1)(1-\sigma(q_1))] * 1 \\
&= \begin{bmatrix} 3.67 * 10^{-4} \\ \\ 1.71 * 10^{-5} \end{bmatrix}
\end{aligned} \tag{14}
$$

3. Update parameters.

$$
b_y = b_y - 0.01 * \frac{\partial L}{\partial b_y} \quad = 1 - 0.01 * 0.0069 = 0.999931 \quad = \begin{bmatrix} 3.67 * 10^{-4} \\ \\ 1.71 * 10^{-5} \end{bmatrix} \tag{15}
$$

$$
\begin{aligned}
W_h &= W_h - 0.01 * \frac{\partial L}{\partial W_h} \\
&= \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} - 0.01 * \begin{bmatrix} 7.37 * 10^{-4} & 3.43 * 10^{-5} \\ 3.67 * 10^{-4} & 1.71 * 10^{-5} \end{bmatrix} \\
&= \begin{bmatrix} 0.99999263 & 2.999999657 \\ 1.99999633 & 0.999999829 \end{bmatrix}
\end{aligned} \tag{16}
$$

3. **Solution to Problem 2.3**

   1. My context free grammar:

   S $--> $ NP+VP

   PP $--> $ ADP + NP | ADP + N

   NP $--> $ DET + NOUN | NP + PP | DET + NN | DET + ADJP | NP + CCONJP | NN

   VP $--> $ V+ADVP | V + NP

   ADJP $--> $ ADJ+NOUN | ADJ + NP | ADJ+NN

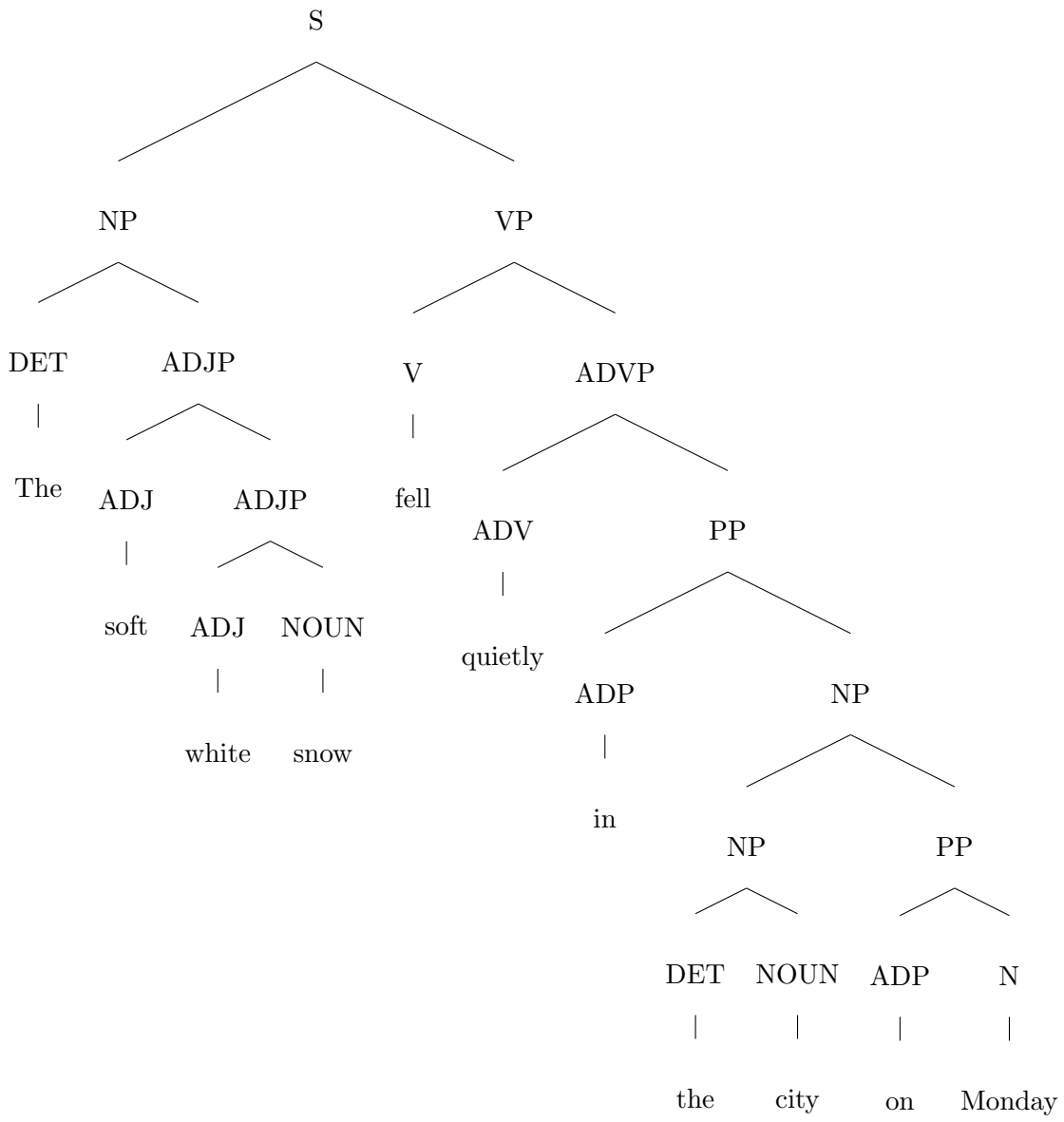
   ADVP $--> $ ADV + PP | ADV+ADJP

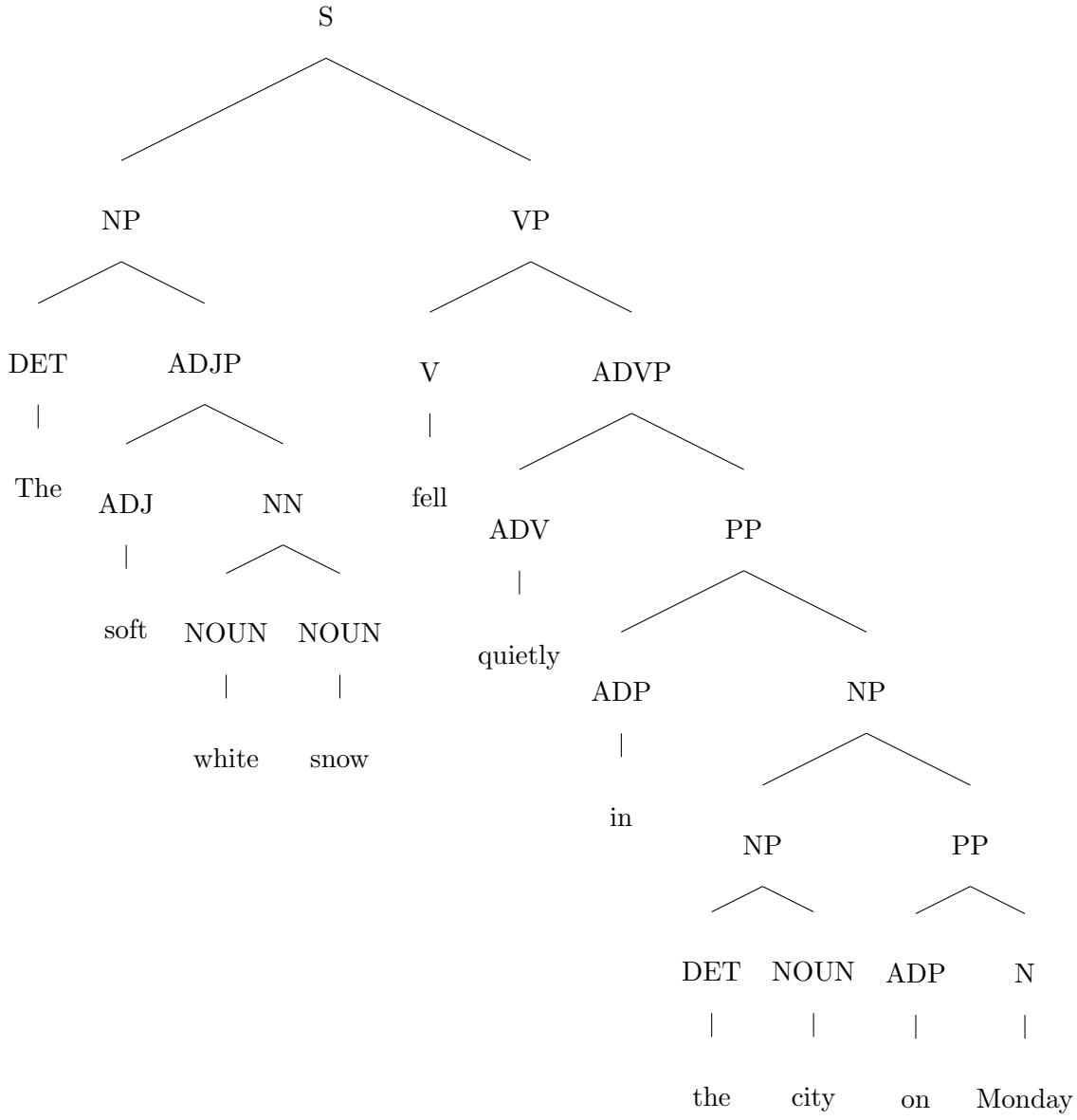
   CCONJP $--> $ CCONJ + S

   NN $--> $ NOUN+NOUN

   There are 18 rules in my Context Free Grammar.

   2. There are two possible parse trees for the sentence (a) using my CFG.


S
NP
VP
DET
ADJP
V
ADVP
The
ADJ
ADJP
fell
ADV
PP
soft
ADJ
NOUN
quietly
ADP
NP
white
snow
in
NP
PP
DET
NOUN
ADP
N
the
city
on
Monday



4. **Solution to Problem 2.4**
   1. I will not show all the connections in my graph just to make the graph clear.

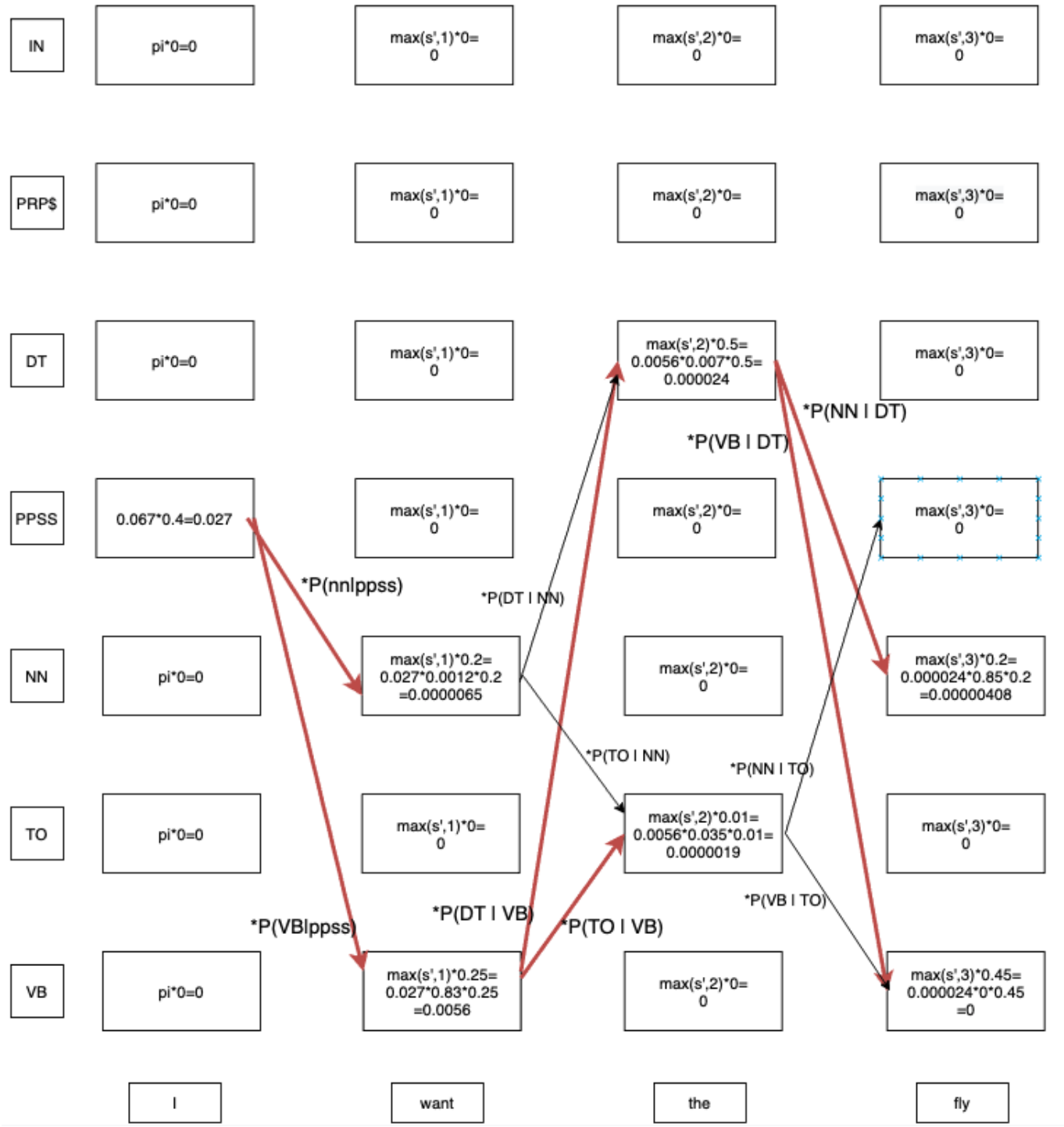


Figure 2: Figure for the dynamic programming trellis

2. From the above dynamic programming trellis, we can use back tracing to get the best path for fly as a noun or as a verb. When fly is a noun:

$$\begin{aligned} v_4 &= P(\text{PPSS}| < START >) * P(\text{I}|PPSS) * P(\text{VB}|PPSS) \\ &\quad * P(\text{want}|VB) * P(\text{DT}|VB) * P(\text{the}|DT) \\ &\quad * P(\text{NN}|DT) * P(\text{fly}|NN) \end{aligned} \tag{17}$$

When fly is a verb:

$$\begin{aligned} v_4 &= P(\text{PPSS}| < START >) * P(\text{I}|PPSS) * P(\text{VB}|PPSS) \\ &\quad * P(\text{want}|VB) * P(\text{DT}|VB) * P(\text{the}|DT) \\ &\quad * P(\text{NN}|DT) * P(\text{fly}|VB) \\ &\quad = 0 \end{aligned} \tag{18}$$

Since $P(\text{fly}|VB)$=0, we know fly here should be used as a noun.