COMS 4705 NLP HW2

Kate Harwood: krh2154@columbia.edu

October 2021

Using 1 late day

PROBLEM 2.1 CODING REFLECTION

Extension1

For my first extension, I implemented custom learning rate scheduling. I first implemented a simple learning rate scheduler, based off of how pytorch's built in learning rate scheduler is implemented. Then I implemented a more involved learning rate scheduler, based off of one of the explanations of learning rate scheduling in this paper:
https://www.mdpi.com/2073-8994/12/4/660/pdf

I chose the exponential step method from this paper. I implemented these schedulers in a separate train function in hw2.py (called train_model_with_learning_rate_scheduler). Learning rate scheduling should improve performance because, while having a larger learning rate at the beginning of training helps to avoid potential local minima, as we get closer to the true minimum of the objective function, we should be making smaller updates to our weights so we don't overshoot the minimum.

This approach did improve the F1 score of the dense model slightly.

Extension2

For my second extension, I implemented a convolutional network. The network has a 1 dimension convolutional layer to fit the text sequences as well as a dropout layer to avoid overfitting. This extension is in models.py and is named ExperimentalNetwork2. A CNN could have a slight advantage over the RNN, since RNNs are better suited to predicting the next part of a sequence, while CNNs function by creating feature maps for salient portions of the text, which helps in a classification task (and is largely how humans classify text as well).

The convolutional network got the best F1 score of the models.

The dense model gets an F1 score of 0.45.

The RNN model gets an F1 score of 0.47.

The dense model with custom learning rate scheduling (extension1) gets an F1 score of 0.46.

The CNN (extension2) gets an F1 score of 0.49.

Extra:
As an extra experiment, I also tried learning the word embeddings from scratch during the network training (not initializing the embedding layer with the pre-trained embeddings) to see the effect. The overall accuracy was mostly the same, but the network took longer to converge. I ultimately had to switch this back to using pretrained embeddings because of the time constraints for our submission.

PROBLEM 2.2 BACKPROPAGATION WITH RNNS

PROBLEM 2.2.1 FORWARD PROPAGATION

Part 2.2.1.1: Outputs and hidden states on forward pass

Part A:

$$h_{t1} = f(q_{t1})$$

$$q_{t1} = W_x^T x_{t1} + W_h^T h_0 + b_h$$

$$q_{t1} = -4 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$q_{t1} = \begin{bmatrix} -4 \\ -12 \end{bmatrix} + \begin{bmatrix} 4 \\ 7 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$q_{t1} = \begin{bmatrix} 1 \\ -3 \end{bmatrix}$$

$$f(q_{t1}) = f(\begin{bmatrix} 1 \\ -3 \end{bmatrix})$$

$$f(q_{t1}) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\boxed{h_{t1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}}$$

Part B:

$$\hat{y}_{t1} = f(p_{t1})$$

$$p_{t1} = W_y^T h_{t1} + b_y$$

We can plug in our value from h_{t1} from above:

$$p_{t1} = \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + [1]$$

$$p_{t1} = 4$$

$$f(p_{t1}) = 4$$

$$\boxed{\hat{y}_{t1} = 4}$$

Part C:

$$h_{t2} = f(q_{t2})$$

$$q_{t2} = W_x^T x_{t2} + W_h^T h_{t1} + b_h$$

We can plug in our value from h_{t1} from above:

$$q_{t2} = 1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$q_{t2} = 1 \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$q_{t2} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$$

$$f(q_{t2}) = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$$

$$\boxed{h_{t2} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}}$$

Part D:

$$\hat{y}_{t2} = f(p_{t2})$$

$$p_{t2} = W_y^T h_{t2} + b_y$$

We can plug in our value from h_{t2} from above:

$$p_{t2} = \begin{bmatrix} 3 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 8 \end{bmatrix} + [1]$$

$$p_{t2} = 18$$

$$f(p_{t2}) = 18$$

$$\boxed{\hat{y}_{t2} = 18}$$

Part 2.2.1.2: MSE

$$L_{MSE} = \frac{1}{N} \Sigma_{i=1}^{n} \Sigma_{t=1}^{m_i} (y_t^i - \hat{y}_t^i)^2$$

Because in our case $N = 1$ we can rewrite this as:

$$L_{MSE} = \Sigma_{t=1}^{m_i} (y_t^i - \hat{y}_t^i)^2$$

We have two timesteps, so we can rewrite as:

$$L_{MSE} = (y_t^1 - \hat{y}_t^1)^2 + (y_t^2 - \hat{y}_t^2)^2$$

We can plug in our known values for y_t^1 and y_t^2 and our values from $\hat{y}_t^1$ and $\hat{y}_t^2$from above:

$$L_{MSE} = (10-4)^2 + (20-18)^2$$

$$L_{MSE} = 36 + 4$$

$$\boxed{L_{MSE} = 40}$$

PROBLEM 2.2.2 BACKPROPAGATION THROUGH TIME

1: Computation Graph

Shown on attached paper.

2: Gradient Calculations

Because N = 1 in this case:

$$L_{MSE} = (20-y_2)^2 + (10-y_1)^2$$

We can start by defining the partial derivatives we will need in these calculations, using our values from the forward propagation above to plug in where we can:

$$\frac{\partial MSE}{\partial y_1} = 2(20-y_1)(-1)$$

$$\frac{\partial MSE}{\partial y_2} = 2(10-y_2)(-1)$$

$$\frac{\partial y_1}{\partial p_{t1}} = 1$$

$$\frac{\partial y_2}{\partial p_{t2}} = 1$$

$$\frac{\partial p_{t1}}{\partial W_y} = h_{t1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\frac{\partial p_{t2}}{\partial W_y} = h_{t2} = \begin{bmatrix} 3 \\ 8 \end{bmatrix}$$

$$\frac{\partial p_{t1}}{\partial b_y} = 1$$

$$\frac{\partial p_{t2}}{\partial b_y} = 1$$

$$\frac{\partial p_{t1}}{\partial h_{t1}} = W_y^T = \begin{bmatrix} 3 & 1 \end{bmatrix}$$

$$\frac{\partial p_{t2}}{\partial h_{t2}} = W_y^T = \begin{bmatrix} 3 & 1 \end{bmatrix}$$

Here we show the Jacobian matrix for the following derivatives:

$$\frac{\partial h_{t1}}{\partial q_{t1}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\frac{\partial h_{t2}}{\partial q_{t2}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\frac{\partial q_{t1}}{\partial W_x} = \begin{bmatrix} -4 & 0 \\ 0 & -4 \end{bmatrix}$$

For the next one we need to use the product rule:

$$\frac{\partial q_{t2}}{\partial W_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + W_h^T \frac{\partial h_{t1}}{\partial q_{t1}} \frac{\partial q_{t1}}{\partial W_x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -4 & 0 \\ 0 & -4 \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ -12 & 1 \end{bmatrix}$$

The next two involve 3D tensors; I have written them in two "layers" using numerator format for each.
For any q_{tx}, the first layer corresponds to the derivatives wrt the first term in q_{tx} and the second layer corresponds to the derivatives wrt to the second term in q_{tx}:

$$\frac{\partial q_{tx}}{\partial W_h} = Layer1 : \begin{bmatrix} \frac{\partial q_{tx}1}{\partial W_{11}} & \frac{\partial q_{tx}1}{\partial W_{21}} \\ \frac{\partial q_{tx}1}{\partial W_{12}} & \frac{\partial q_{tx}1}{\partial W_{22}} \end{bmatrix} Layer2 : \begin{bmatrix} \frac{\partial q_{tx}2}{\partial W_{11}} & \frac{\partial q_{tx}2}{\partial W_{21}} \\ \frac{\partial q_{tx}2}{\partial W_{12}} & \frac{\partial q_{tx}2}{\partial W_{22}} \end{bmatrix}$$

So for $\frac{\partial q_{t1}}{\partial W_h}$:

$$Layer1 = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix} Layer2 = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix}$$

$\frac{\partial q_{t2}}{\partial W_h}$ needs to use the product rule. For first term from the product rule we can do the as above to get: TERM1 $\frac{\partial q_{t2}}{\partial W_h} =$

$$Layer1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} Layer2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

TERM2 $\frac{\partial q_{t2}}{\partial W_h}$ =

$$W_h^T \frac{\partial h_{t1}}{\partial q_{t1}} \frac{\partial q_{t1}}{\partial W_h}$$

$$\frac{\partial q_{t2}}{\partial W_h} = TERM1 + TERM2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + W_h^T \frac{\partial h_{t1}}{\partial q_{t1}} \frac{\partial q_{t1}}{\partial W_h}$$

And finally we have:

$$\frac{\partial q_{t1}}{\partial b_h} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\frac{\partial q_{t2}}{\partial b_h} = W_h^T \frac{\partial h_{t1}}{\partial q_{t1}} \frac{\partial q_{t1}}{\partial b_h} + \frac{\partial q_{t1}}{\partial b_h} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix}$$

Part A: $\frac{\partial MSE}{\partial W_y}$

Using the chain rule:

$$\frac{\partial MSE}{\partial W_y} = \frac{\partial MSE}{\partial y_1} \frac{\partial y_1}{\partial p_{t1}} \frac{\partial p_{t1}}{\partial W_y} + \frac{\partial MSE}{\partial y_2} \frac{\partial y_2}{\partial p_{t2}} \frac{\partial p_{t2}}{\partial W_y}$$

Plugging in the values for these partial derivatives as defined above we have:

$$2(20-18)(-1)\cdot(1)\cdot\begin{bmatrix}3\\8\end{bmatrix} + 2(10-4)(-1)\cdot(1)\cdot\begin{bmatrix}1\\0\end{bmatrix}$$

$$= -4\begin{bmatrix}3\\8\end{bmatrix} - 12\begin{bmatrix}1\\0\end{bmatrix}$$

$$\frac{\partial MSE}{\partial W_y} = \boxed{\begin{bmatrix}-24\\-32\end{bmatrix}}$$

Part B: $\frac{\partial MSE}{\partial W_h}$

$$\frac{\partial MSE}{\partial W_h} = \frac{\partial MSE}{\partial y_1} \frac{\partial y_1}{\partial p_{t1}} \frac{\partial p_{t1}}{\partial h_{t1}} \frac{\partial h_{t1}}{\partial q_{t1}} \frac{\partial q_{t1}}{\partial W_h} + \frac{\partial MSE}{\partial y_2} \frac{\partial y_2}{\partial p_{t2}} \frac{\partial p_{t2}}{\partial h_{t2}} \frac{\partial h_{t2}}{\partial q_{t2}} \frac{\partial q_{t2}}{\partial W_h}$$

Plugging in our values for the partial derivatives as defined above we get the following. Note I have shown both layers of the 3D tensor derivatives.

$$2(10-4)(-1)\cdot(1)\cdot 1\cdot\begin{bmatrix}3 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 1 & 0\\ 0 & 2\\ 0 & 1\end{bmatrix}+2(20-18)(-1)\cdot 1\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\left(\begin{bmatrix}1 & 2\\ 3 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 1 & 0\\ 0 & 2\\ 0 & 1\end{bmatrix}+\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}\right)$$

$$=\begin{bmatrix}-36 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 1 & 0\\ 0 & 2\\ 0 & 1\end{bmatrix}+\begin{bmatrix}-12 & -4\end{bmatrix}\left(\begin{bmatrix}1 & 2\\ 3 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 1 & 0\\ 0 & 2\\ 0 & 1\end{bmatrix}+\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}\right)$$

$$=\begin{bmatrix}-36 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 1 & 0\\ 0 & 2\\ 0 & 1\end{bmatrix}+\begin{bmatrix}-24 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 1 & 0\\ 0 & 2\\ 0 & 1\end{bmatrix}+\begin{bmatrix}-12 & -4\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}$$

When we multiply by the 3D tensor we think of the layers sitting on top of each other, and each spot in the resulting matrix is the product of a vector by a vector:

$$=\begin{bmatrix}-72 & 0\\ -36 & 0\end{bmatrix}+\begin{bmatrix}-48 & 0\\ -24 & 0\end{bmatrix}+\begin{bmatrix}-12 & -4\\ 0 & 0\end{bmatrix}$$

$$\frac{\partial MSE}{\partial W_h}=\boxed{\begin{bmatrix}-132 & -4\\ -60 & 0\end{bmatrix}}$$

Part C: $\frac{\partial MSE}{\partial W_x}$

$$\frac{\partial MSE}{\partial W_x}=\frac{\partial MSE}{\partial y_1}\frac{\partial y_1}{\partial p_{t1}}\frac{\partial p_{t1}}{\partial h_{t1}}\frac{\partial h_{t1}}{\partial q_{t1}}\frac{\partial q_{t1}}{\partial W_x}+\frac{\partial MSE}{\partial y_2}\frac{\partial y_2}{\partial p_{t2}}\frac{\partial p_{t2}}{\partial h_{t2}}\frac{\partial h_{t2}}{\partial q_{t2}}\frac{\partial q_{t2}}{\partial W_x}$$

Plugging in our values for the partial derivatives as defined above we get:

$$2(10-4)(-1)\cdot(1)\cdot\begin{bmatrix}3 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}-4 & 0\\ 0 & -4\end{bmatrix}+2(20-18)(-1)\cdot(1)\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}-3 & 0\\ 0 & 1\end{bmatrix}$$

$$=\begin{bmatrix}-36 & -12\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}-4 & 0\\ 0 & -4\end{bmatrix}+\begin{bmatrix}-12 & -4\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}-3 & -12\\ 0 & 1\end{bmatrix}$$

$$= \begin{bmatrix}144 & 0\end{bmatrix} + \begin{bmatrix}84 & -4\end{bmatrix}$$
$$\frac{\partial MSE}{\partial W_x} = \boxed{\begin{bmatrix}228 & -4\end{bmatrix}}$$

Part D: $\frac{\partial MSE}{\partial b_y}$

$$\frac{\partial MSE}{\partial b_y} = \frac{\partial MSE}{\partial y_1}\frac{\partial y_1}{\partial p_{t1}}\frac{\partial p_{t1}}{\partial b_y} + \frac{\partial MSE}{\partial y_2}\frac{\partial y_2}{\partial p_{t2}}\frac{\partial p_{t2}}{\partial b_y}$$

$$2(20-18)(-1)\cdot(1) + 2(10-4)(-1)\cdot(1)$$
$$= -4 - 12$$
$$\frac{\partial MSE}{\partial b_y} = \boxed{-16}$$

Part E: $\frac{\partial MSE}{\partial b_h}$

$$\frac{\partial MSE}{\partial b_h} = \frac{\partial MSE}{\partial y_1}\frac{\partial y_1}{\partial p_{t1}}\frac{\partial p_{t1}}{\partial h_{t1}}\frac{\partial h_{t1}}{\partial q_{t1}}\frac{\partial q_{t1}}{\partial b_h} + \frac{\partial MSE}{\partial y_2}\frac{\partial y_2}{\partial p_{t2}}\frac{\partial p_{t2}}{\partial h_{t2}}\frac{\partial h_{t2}}{\partial q_{t2}}\frac{\partial q_{t2}}{\partial b_h}$$

Plugging in our values for the partial derivatives as defined above we get:

$$2(10-4)(-1)\cdot(1)\cdot\begin{bmatrix}3 & 1\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + 2(20-18)(-1)\cdot(1)\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}2 & 0\\ 3 & 1\end{bmatrix}$$
$$= \begin{bmatrix}-36 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + \begin{bmatrix}-12 & -4\end{bmatrix}\begin{bmatrix}2 & 0\\ 3 & 1\end{bmatrix}$$
$$\frac{\partial MSE}{\partial b_h} = \boxed{\begin{bmatrix}-72 & -4\end{bmatrix}}$$

Problem 3: Updating Weights

Part a

$$b_y = b_y - 0.01(\frac{\partial MSE}{\partial b_y})$$
$$b_y = b_y - 0.01(-16) = \boxed{1.16}$$

Part b

$$W_h = W_h - 0.01(\frac{\partial MSE}{\partial W_h})$$

$$W_h = W_h - 0.01(\begin{bmatrix} -132 & -4 \\ -60 & 0 \end{bmatrix})$$

$$= \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} -1.32 & -0.04 \\ -0.6 & 0 \end{bmatrix}$$

$$= \boxed{\begin{bmatrix} 2.32 & 3.04 \\ 2.6 & 1 \end{bmatrix}}$$

PROBLEM 2.3 SYNTAX AND GRAMMAR

Problem 2.3.1 Parse Trees

Part 1: Context Free Grammar
Boots were not made for sitting by the door.
You do not want to stay anymore.
You can have your space, cowboy.
I am not going to fence you in.

Rules:
1. PP → P NP

2. PP → P VP

3. NP → D N

4. NP → Pron N

5. VP → VP VP

6. VP → V ADV

7. VP → V PP

8. VP → V NP

9. VP → MV VP

10. VP → V Pron ADV

11. S → Pron VP N

Part 2: Parse Trees
Shown on attached pages.

Problem 2.3.2 Viterbi Algorithm

Part 1: Dynamic Programming Trellis

Shown on a attached page. I've included only the transitions for the non-zero observation likelihoods so the trellis is more readable. The "max" paths (paths we can backtrace to get the formula) are slightly highlighted with double arrows.

Part 2: Formulas

$P(burn = N|sentence) = P(N|start) * P(It|N) * P(V|N) * P(is|V) * P(D|V) * P(a|D) * P(A|D) * P(slow|A) * P(N|A) * P(burn|N) =$

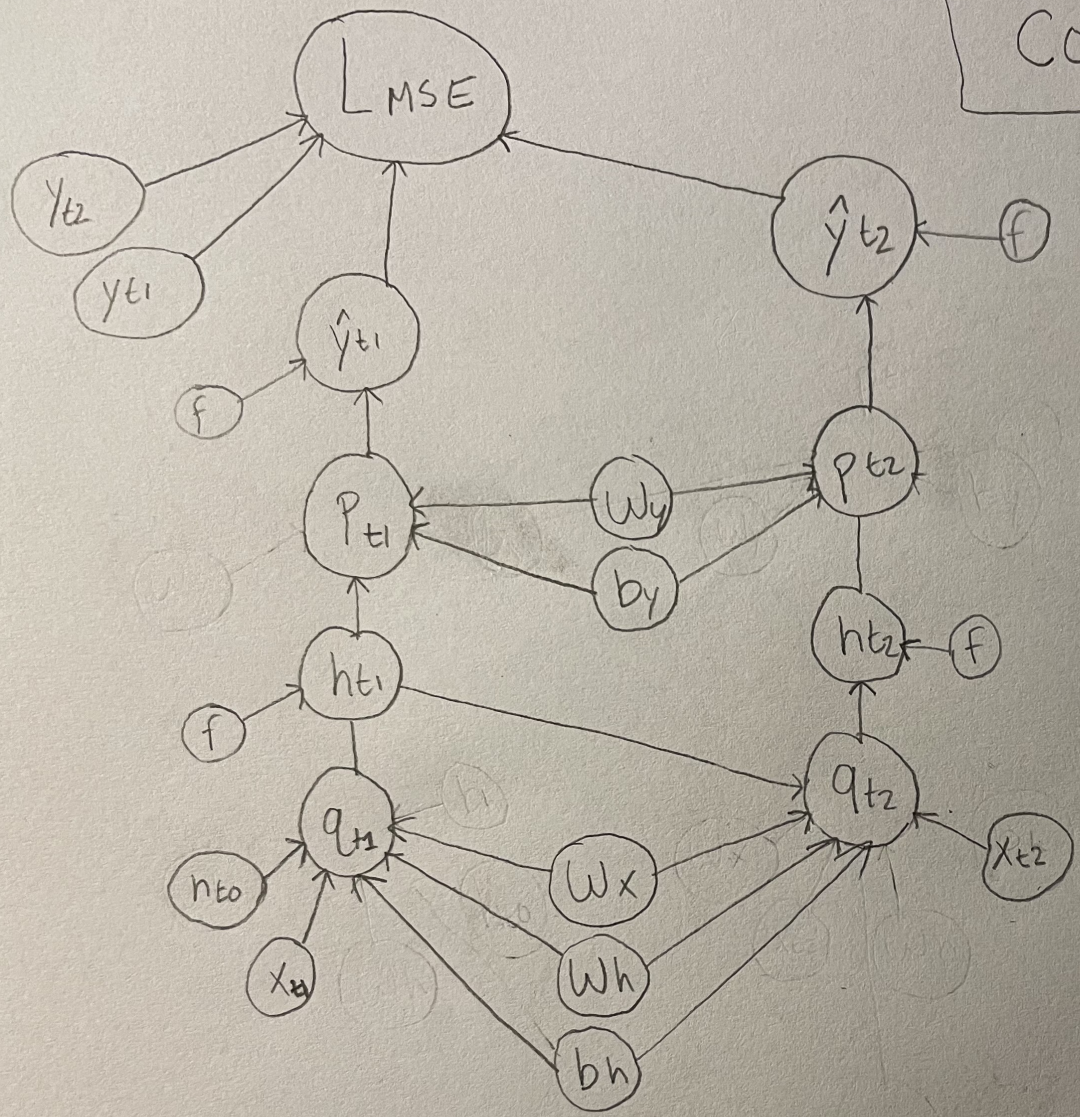
$(.6) * (.4) * (.8) * (.8) * (.8) * (1) * (.5) * (1) * (.8) * (.5)$
$= .024576$

$P(burn = V|sentence) = P(N|start) * P(It|N) * P(V|N) * P(is|V) * P(N|V) * P(a|N) * P(V|N) * P(slow|V) * P(V|V) * P(burn|V) =$

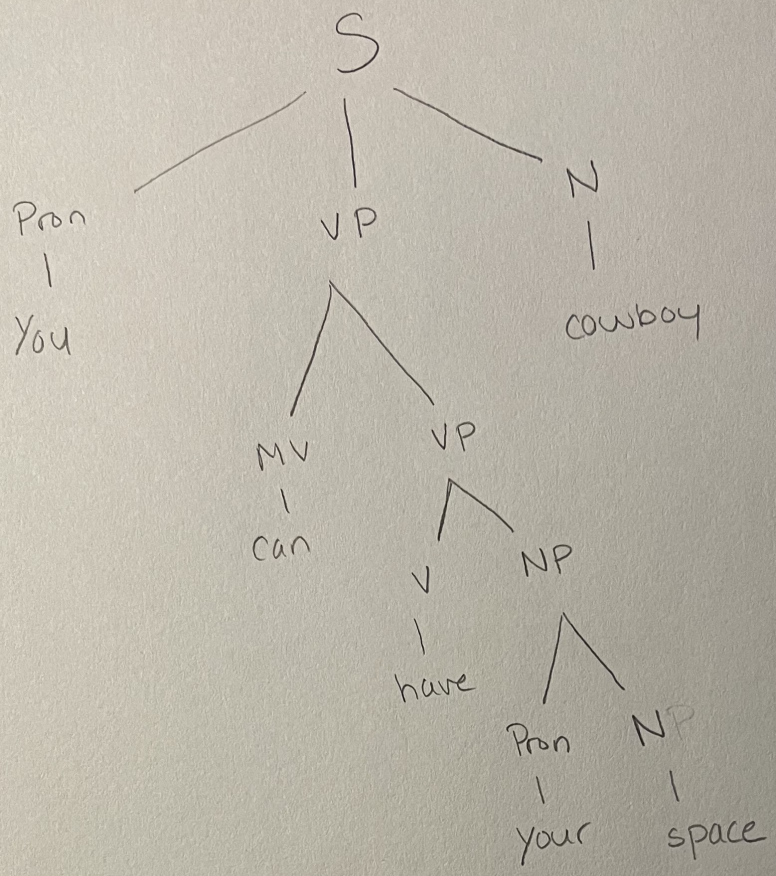
$(.6) * (.4) * (.8) * (.8) * (.05) * (.1) * (.8) * (.05) * (.05) * (.05)$
$= 7.68e{-}8$

Computation Graph

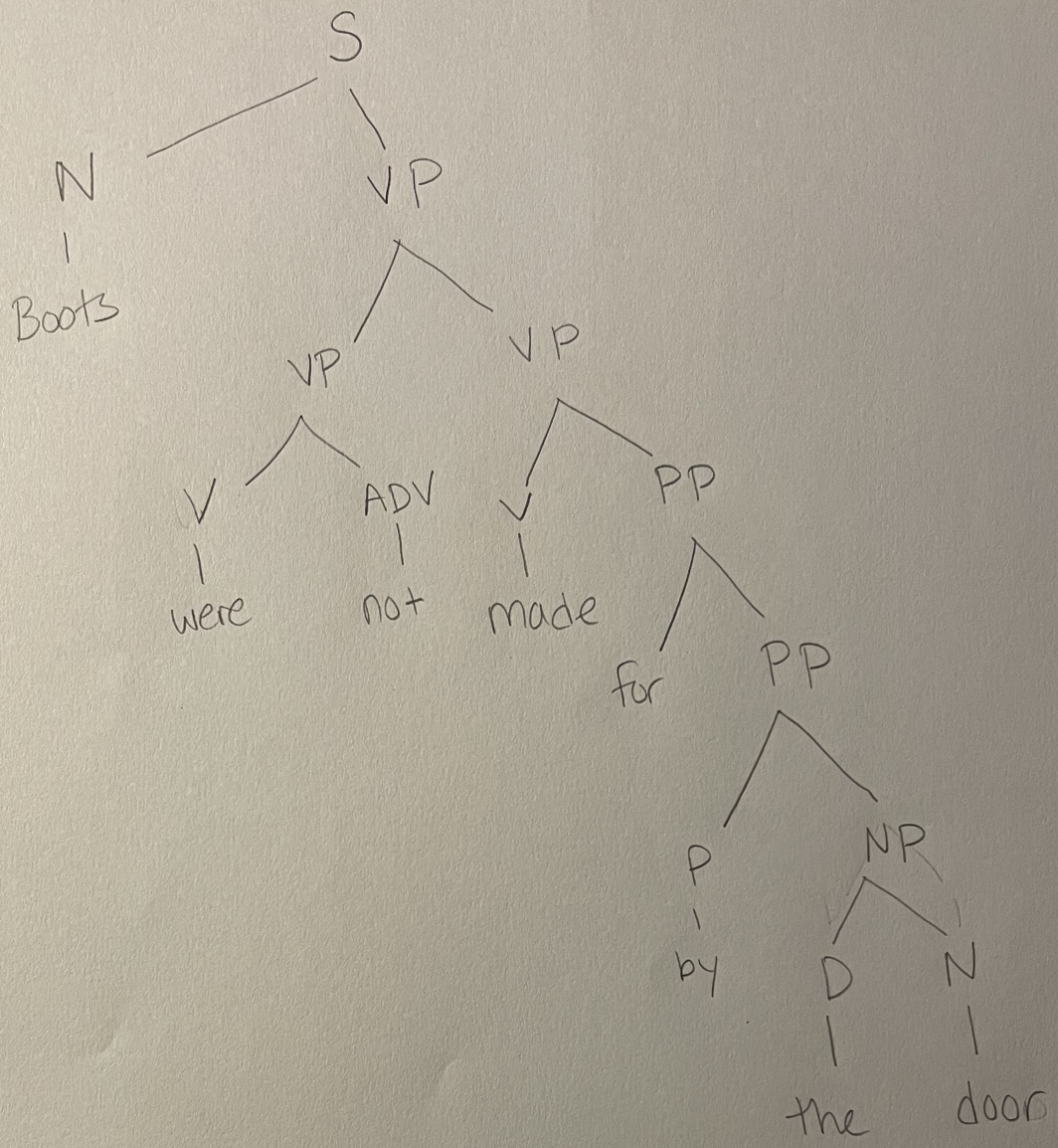
krh2154



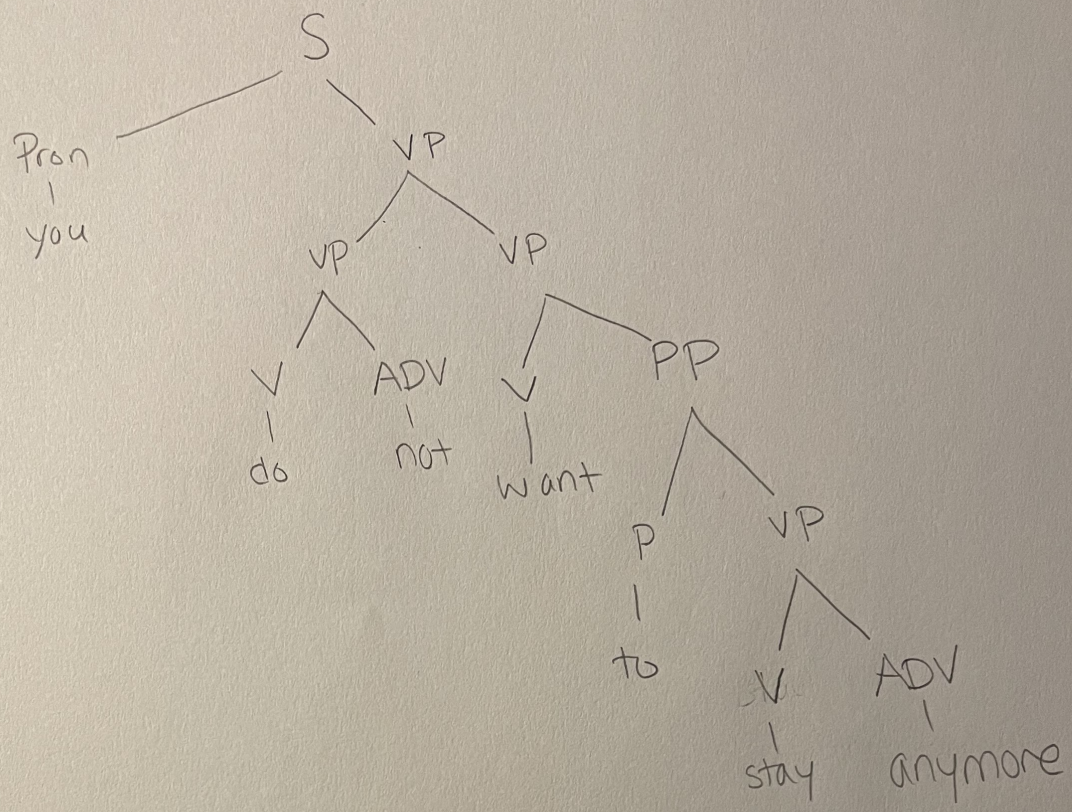
Krh2154



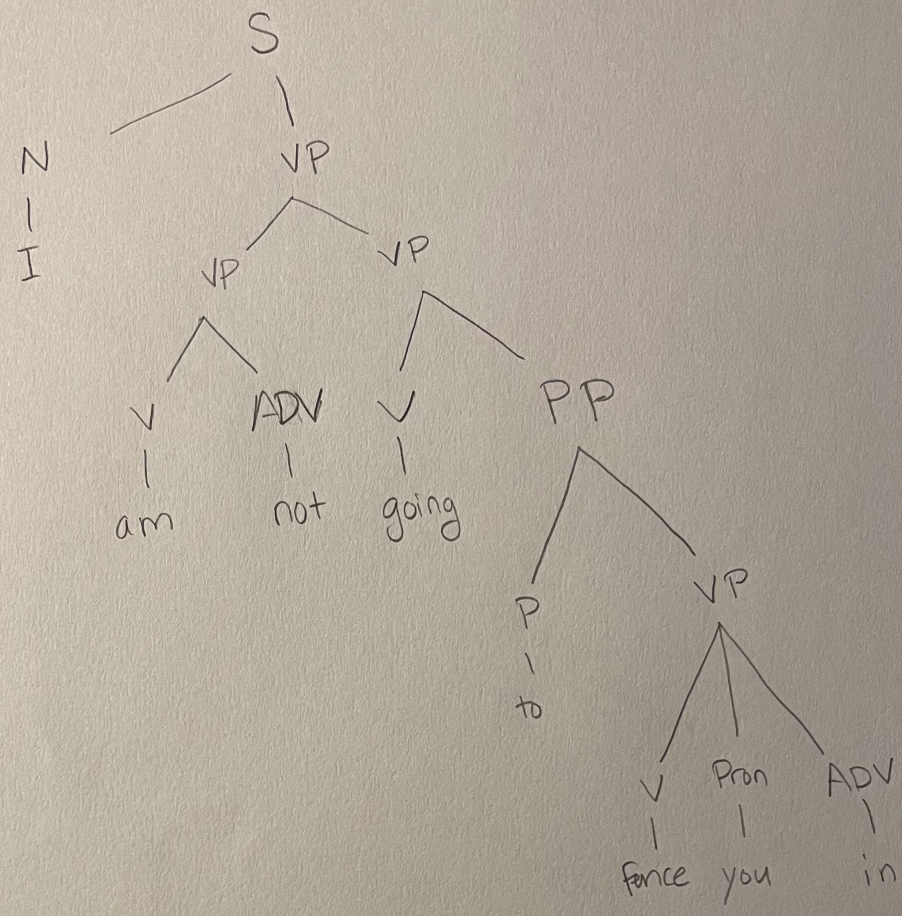
krh2154

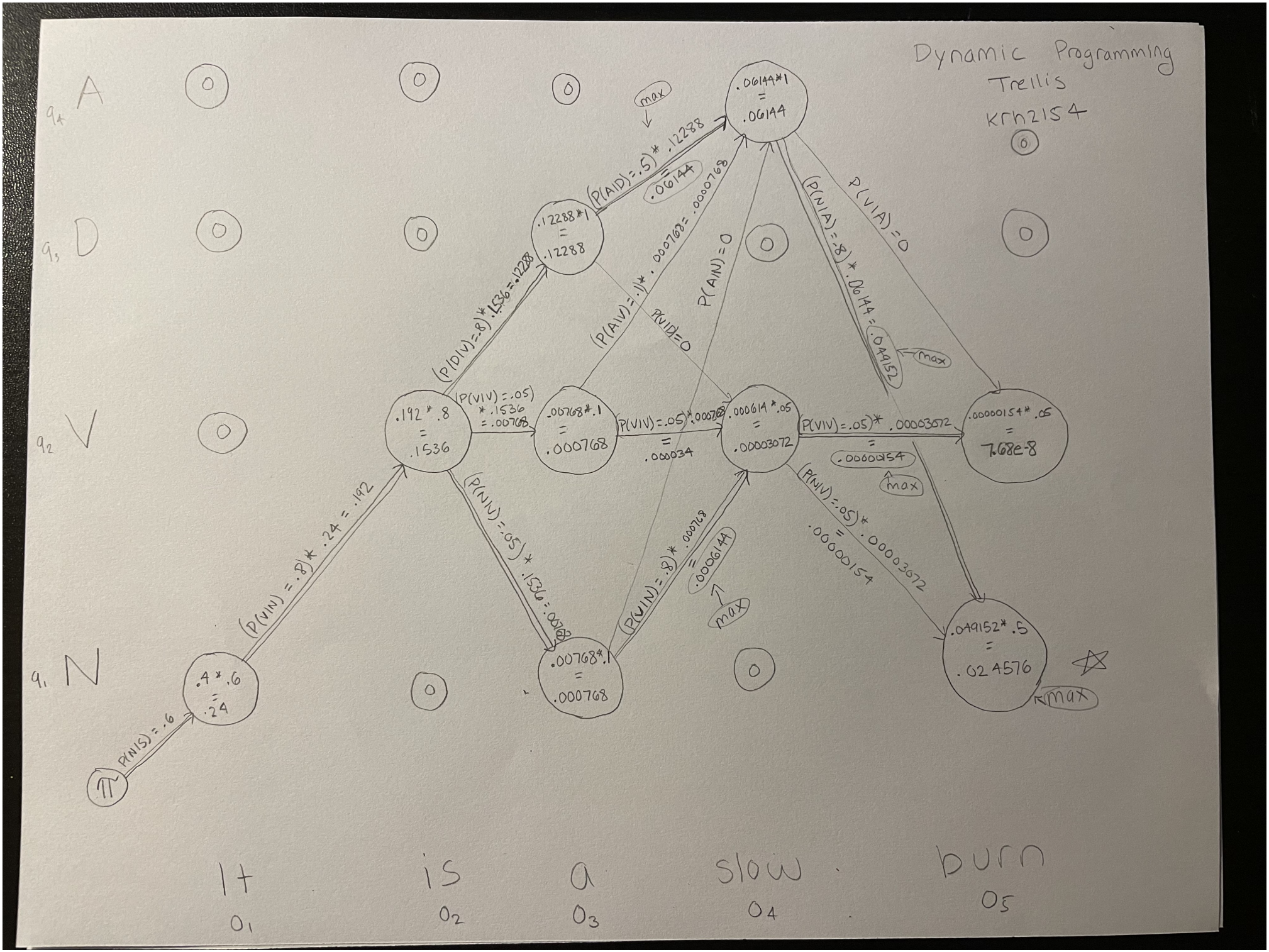

S
N
Boots
VP
VP
V
were
ADV
not
VP
V
made
PP
for
PP
P
by
NP
D
the
N
door

krh2154


S
Pron
you
VP
VP
VP
V
do
ADV
not
V
want
PP
P
to
VP
V
stay
ADV
anymore

krh2154




Dynamic Programming
Trellis
krh2154
q1 A
q3 D
q2 V
q1 N
π
P(N|S) = .6
.4 * .6 = .24
(P(V|N) = .8) * .24 = .192
.192 * .8 = .1536
(P(D|V) = .8) * .1536 = .1228
.12288 * 1 = .12288
(P(A|D) = .5) * .12288 = .06144
.06144 * 1 = .06144
(P(V|V) = .05) * .1536 = .00768
.00768 * .1 = .000768
(P(A|V) = .1) * .000768 = .0000768
P(V|D) = 0
P(A|N) = 0
(P(V|N) = .8) * .00768 = .006144
.006144 * .1 = .000614
(P(V|V) = .05) * .000768 = .000034
.000614 * .05 = .00003072
(P(N|A) = .8) * .06144 = .049152
.049152 * .5 = .024576
P(V|A) = 0
(P(V|V) = .05) * .00003072 = .00000154
.00000154 * .05 = 7.68e-8
(P(N|V) = .05) * .00003072 = .00000154
max
It
is
a
slow
burn
o1
o2
o3
o4
o5