# CPSC 477/577 Natural Language Processing – Spring 2021

**Homework 3**

**Due 4/19/21, 11:59:59 PM**

*Instructor: Dragomir Radev*
*HW3 TA in charge: Aarohi Srivastava*

**Instructions:**

- Please submit a legible PDF for this assignment on Canvas by the due date.
- This document contains 10 questions. Please select 8 of these 10 to solve. **Do not** put more than 8 questions in your final submission - we will not count extra credit, nor will we select your best scoring problems.
- The assignment will be scored out of 80, with each of the 8 questions you choose worth 10 points.
- Your submission can include handwritten and/or typed work, as long as you show all your work and clearly mark your final answer for each part. Partial credit will be given only if work is shown clearly.

**1. Probability**

Assume a bigram language model that is trained on the sentences below. Ignore P(<s>) of the start token when estimating the unigram model.

<s> dog chases cat </s>
<s> cat chases dog </s>
<s> cat chases cat </s>
<s> dog bites cat </s>
<s> cat bites dog </s>

Part 1: What is the unigram probability of *dog*? (Round to the nearest 0.01, if necessary.)

Part 2: What is the bigram probability of *chases dog*? (Round to the nearest 0.01, if necessary.)

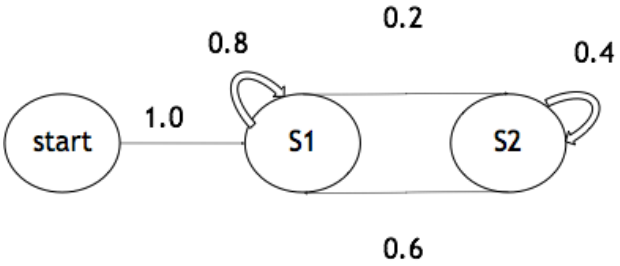
Part 3: What is the estimated probability of the test sentence below?

<s> dog chases dog </s>

Use MLE with linear interpolation for smoothing (set the bigram weight to 0.8 and the unigram weight to 0.2). (Round to the nearest 0.0001, if necessary.)

## 2. Hidden Markov Model

Consider the following HMM, where the state transition probabilities are shown on the arrows:




The emission probabilities are shown in the following table.
For example, p(a|S1) = 0.2.

| | c | a | t |
|---|---|---|---|
| S1 | 0.7 | 0.2 | 0.1 |
| S2 | 0.3 | 0.5 | 0.2 |

What is the probability of the observation sequence “at”?

What is the probability of the observation sequence “act”?

**3. N-Gram Models**

Consider the sentence

> "Bats (V/N) are (V) cool (V/N/Adj) . (STOP)".

(a) Consider a *bigram* HMM model for tagging this sentence. You are given the following transition and emission log probabilities.

Transition (q):

> log P(V | *) = log P(N | *) = log P(Adj | *) = -0.1
> log P(V | V) = -2; log P(V | N) = -0.05
> log P(N | V) = -0.3; log P(Adj | V) = -0.7
> log P(STOP | V) = log P(STOP | N) = -5; log P(STOP | Adj) = -6

Emission (e)

> log P("Bats" | V) = log P("Bats" | N) = -6
> log P("are" | V) = -4
> log P("cool" | V) = -7; log P("cool" | N) = -8; log P("cool" | Adj) = -4
> log P("." | STOP) = 0

Recall that the Viterbi algorithm uses a DP table, with $\pi(k, v)$ indicating the maximum log probability of observing tag $v$ at position k. For a bigram tagger, the base case and recursive case are:

> $\pi(0, \text{'*'}) = 0$
> $\pi(k, v) = \max_{u \in K} (\pi(k-1, u) + q(v|u) + e(word\_k|v))$

where $K$ is the space of all tags.

Complete the DP table as necessary to obtain probabilities for all possible tag sequences. Cells with 0 probability can be left blank. Circle entries constituting the most probable path. Show work for partial credit.

| | k = 1 | k = 2 | k = 3 | k = 4 |
|---|---|---|---|---|
| *v* | Bats (V/N) | are (V) | cool (V/N/Adj) | . (STOP) |
| V | | | | |
| N | | | | |

| | | | | |
|---|---|---|---|---|
| Adj | | | | |
| STOP | | | | |

What is the most probable tag sequence and its log probability?

Word: Bats are cool .

Best tag: ____ ____ ____ ____

Log probability of this tag sequence = _____

(b) In one sentence, explain why a trigram HMM model might be preferred to a bigram model.

**4. Training a Neural Network**

This question is to train a three-layer neural network for classification task. Let $H1$ denote the number of hidden units, let $D$ be the dimension of the input $X$, and let $K$ be the number of classes. The three-layer network has the following form:

$$h_1 = ReLU(W_1 X + b_1)$$
$$h_2 = ReLU(W_2 h_1 + b_2)$$
$$p = Softmax(W_3 h_2 + b_3)$$

where the parameters of the network are $W_1 \in R^{H_1 * D}$, $b_1 \in R^{H_1}$, $W_2 \in R^{H_2 * H_1}$, $b_2 \in R^{H_2}$, $W_3 \in R^{K * H_2}$, $b_3 \in R^K$.

$ReLU$ is the "rectified linear unit" $ReLU(x) = max\{0, x\}$ applied component-wise, and Softmax maps a d-vector to the probability simplex according to:

$$Softmax(v)_k = \frac{exp(v_k)}{\sum_{j=1}^{K} exp(v_j)}$$

For a given input $X$, this specifies how to calculate the probabilities $P(Y = k|X)$ for $k = 1,2,\ldots,k$.

For a given training set $\{(X_i, Y_i)\}$, consider the loss function:

$$L = \frac{1}{n}\sum_{i=1}^{n} - log\, p(Y = Y_i|X_i) + \frac{\lambda}{2}(\|W_1\|^2 + \|W_2\|^2 + \|W_3\|^2)$$

where $\|A\|^2$ means the sum of the squares of the entries of the matrix $A$.

Give formulas for the derivatives for each layer of the network$j = 1,2,3$.

$$\frac{\partial L}{\partial W_j}, \frac{\partial L}{\partial b_j}$$

Hints: You should ignore the fact that $ReLU(x)$ is not differentiable at $x_i = 0$. If needed, you might also try to first do the calculations by replacing $ReLU$ with the identity function.

**5. Word Embeddings**

A. Consider the following word embeddings: a for “America”, c for “China” and w for “Washington DC”. Write an expression approximating b (“Beijing”, the capital of China) in terms of a, c, and w.

B. What does it mean for a word w1 to be a hypernym of w2? Give your definition in terms of [[w1]], the set of referents that “are a” w1, and [[w2]], the set of referents that “are a” w2.

C. If w1 is a hypernym of w2, must the linguistic distribution of w1 be similar to the linguistic distribution of w2? Give a heuristic argument using examples in English (or another language).

D. Must it be the case that a word and its hypernym will have similar word2vec vectors? Hint: Think about the assumptions built into the training process for word2vec embeddings.

**6. Sentence Similarity**

Consider the following three documents,

D1 = "dogs chase cats"
D2 = "I love cats and dogs"
D3 = "I love computer science"

Recall the relevant equations:

$$Jaccard(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \quad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \quad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$\text{PPMI}_{ij} = \max(\log_2 \frac{p_{ij}}{p_{i*}p_{*j}}, 0)$$

(1) What is the Jaccard similarity of D1 and D2 if considering documents as sets of unigrams? You can give your answer in fraction form.

Jaccard (D1,D2) = _____

(2) What is the Jaccard similarity of D2 and D3 if considering documents as sets of bigrams? You can give your answer in fraction form.

Jaccard (D2,D3) = _____

(3) Fill out the term-document matrix below, considering the vocabulary = {dogs, chase, cats, love, computer, science}.

| | dogs | chase | cats | love | computer | science |
|---|---|---|---|---|---|---|
| D1 | | | | | | |
| D2 | | | | | | |
| D3 | | | | | | |

(4) What is the cosine similarity of D1 and D2 based on term-document matrix above?

cosine (D1,D2) = _____

(5) What is the euclidean distance of D2 and D3 based on term-document matrix above?

euclidean (D2,D3) = _____

(6) Fill out the term-term matrix below, considering target words = {chase,love} and context words = {dogs,cats,computer,science}.

| | dogs | cats | computer | science |
|---|---|---|---|---|
| chase | | | | |
| love | | | | |

(7) Calculate the PPMI(w = chase, c = cats) and PPMI(w = chase, c = computer). You can give your answer in log form.

PPMI (w = chase, c = cats) = _____

PPMI (w = chase, c = computer) = _____

**7. Noisy Channel Model**

For your English 120 class, you're asked to write an essay about your favorite animal. You produce the following rough draft.

The cas in the cas were cas cute cas my cas driver's cas.
Cas my cas do not like being in the cas, I do not drive any cas.
From the cas, I watched some cas cas they played with other cas.

You hire an expert spell checker to look over your draft, and they respond with the following, spell-checked piece:

The cat in the car was as cute as my car driver's cat.
My cat does not like being in the car, so I do not drive any cat.
From the car, I watched a cat as it played with another cat.

a. Compute unigram probabilities for the following words based on the spell-checked draft.

P(*cat*) =
P(*car*) =
P(*as)* =

b. Recall the noisy channel model. Fill in the probabilities from a confusion matrix based on the spell checker's edits to your draft.

Substitution:

sub[s, t] =
sub[s, r] =

Insertion:

ins[c, a] =

c. You find you wrote another sentence for your essay that you forgot to include initially.

*The cas is fast.*

You know *cas* isn't a word, and that you certainly meant to write something else. Classic mistake! However, you wrote this sentence a long time ago, so you don't remember exactly what you were trying to say. Use the probabilities you computed in parts (a) and (c) to decide which word you most likely meant to write in place of *cas*. Show all of your work.

## 8. Activation Functions

**Activation functions** Assume a 2-layer feedforward network $\mathbf{y} = f(\mathbf{x})$ such that:

$$\begin{aligned}\mathbf{a} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\ \mathbf{z} &= \sigma(\mathbf{a}) \\ \mathbf{y} &= \mathbf{W}^{(2)}\mathbf{z} + \mathbf{b}^{(2)}\end{aligned}$$

where $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ are the weights and biases for the $i^{th}$ linear layer and $\sigma(\cdot)$ is the sigmoid activation function at the hidden layer. Derive an equivalent network $f^{'}$ — by adjusting $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ — such that $f^{'}(\mathbf{x}) = f(\mathbf{x})$ (i.e both networks produce the same output for a given input) but $f^{'}$ uses the $\tanh(\cdot)$ activation function at the hidden layer.

## 9. Cross Entropy Loss

The distributional hypothesis — popularly stated as "a word is characterized by the company it keeps" — suggests that words that occur in similar contexts should be similar in meaning. The distributional hypothesis forms the basis for the Skipgram model of Mikolov et. al., which is an efficient way of learning the meaning of words as dense vector representations from unstructured text. The skipgram objective is to learn the probability distribution $P(C|T)$ where given a target word $w_t$, we estimate the probability that a context word $w_c$ lies in the context window of $w_t$. The distribution of the probabilistic model is parameterized as follows:

$$P(C = w_c|T = w_t) = \frac{\exp(\mathbf{u}_{w_c}^\top \cdot \mathbf{v}_{w_t})}{\sum_{w' \in \mathcal{V}} \exp(\mathbf{u}_{w'}^\top \cdot \mathbf{v}_{w_t})}, \tag{1}$$

where vectors $\mathbf{u}_{w_c}$ and $\mathbf{v}_{w_t}$ represent the context word $w_c$ and the target word $w_t$ respectively. Notice the use of softmax function and how the problem of learning embeddings in this model has been cast as a classification problem. The vectors for all the words in the vocabulary $\mathcal{V}$ can be succinctly represented in two matrices $\mathbf{U}$ and $\mathbf{V}$, where the vector in the $j^{th}$ column in $\mathbf{U}$ and $\mathbf{V}$ corresponds to the context and target vectors for the $j^{th}$ word in $\mathcal{V}$. Note that $\mathbf{U}$ and $\mathbf{V}$ are the parameters of the model. Answer the following questions about the Skipgram model.

(a) The cross entropy loss between two probability distributions $p$ and $q$, is expressed as:

$$\text{CROSS-ENTROPY}(p, q) = -\sum_m p_m \log(q_m). \tag{2}$$

For a given target word $w_t$, we can consider the ground truth distribution $\mathbf{y}$ to be a one-hot vector of size $|\mathcal{V}|$ with a $1$ only at the true context word $w_c$'s entry, and $0$ everywhere else. The predicted distribution $\hat{\mathbf{y}}$ (same length as $\mathbf{y}$) is the probability distribution $P(C|T = w_t)$. The $j^{th}$ entry in these vectors is the probability of the $j^{th}$ word in $\mathcal{V}$ being a context word. Write a simplified expression of cross entropy loss, $\text{CROSS-ENTROPY}(\mathbf{y}, \hat{\mathbf{y}})$, for the Skipgram model on a single pair of words $w_c$ and $w_t$. Your answer should be in terms of $P(C = w_c|T = w_t)$). [2 pts]

(b) Find the gradient of the cross entropy loss calculated in (a) with respect to the target word vector $\mathbf{v}_{w_t}$. Your answer should be in terms of $\mathbf{y}$, $\hat{\mathbf{y}}$ and $\mathbf{U}$). [4 pts]

(c) Find the gradient of the cross entropy loss calculated in (a) with respect to each of the context word vectors $\mathbf{u}_{w_c}$. Do this for both cases $C = w_c$ (true context word) and $C \neq w_c$ (all other words). Your answer should be in terms of $\mathbf{y}$, $\hat{\mathbf{y}}$ and $\mathbf{v}_{w_t}$. [4 pts]

## 10. Scalar-Valued Variables

**(Scalar-Valued Variables)**: Let $x, w_1, b_1, w_2, b_2 \in \mathbb{R}$ and define

$$
\begin{aligned}
z_1 &= w_1 x \\
z_2 &= z_1 + b_1 \\
z_3 &= \text{ReLU}(z_2) = \max\{0, z_2\} \\
z_4 &= w_2 z_3 \\
z_5 &= z_4 + b_2
\end{aligned}
$$

Thus $z_5 = w_2 \text{ReLU}(w_1 x + b_1) + b_2 \in \mathbb{R}$ is the output of a feedforward network with one nonlinear (ReLU) layer in which all variables are scalars.

(a) Draw the corresponding computation graph with $x, w_1, b_1, w_2, b_2$ as the input nodes, $z_5$ as the output node, and $\times/+/\text{ReLU}$ as internal node types (corresponding to multiplication, addition, and ReLU).

(b) Evaluate $z_5$ with input values $x = 1$, $w_1 = 1/4$, $b_1 = 0$, $w_2 = 1/3$, and $b_2 = 0$ by running the forward pass on the graph.

(c) Run backpropagation to calculate the gradient of $z_5$ with respect to $b_2, w_2, b_1, w_1, x$ evaluated at the input values.

(d) Add the skip connection $z_6 = z_5 + x$ and draw the resulting computation graph (with $z_6$ as the output node). Run the forward pass to evaluate $z_6$ and backpropagation to calculate the gradient of $z_6$ with respect to $b_2, w_2, b_1, w_1, x$ evaluated at the same input values. You only need to do two additional computations (one in forward, one in backward) on top of what you have already done in the previous problem.

(e) What does the gradient with respect to $x$ say about the sensitivity of the function with $x$ before ($z_5$) and after ($z_6$) adding the skip connection?