Midterm for CS371N: Natural Language Processing (Fall 2023)

Instructions:

- You will have 80 minutes to complete the exam.
- This exam is to be completed individually be each student.
- You are allowed one 8.5"x11" double-sided note sheet.
- You are **not** allowed calculators or other electronic devices.
- Partial credit will be given for short-answer and long-answer questions, so it is to your advantage to show work in the exam.
- For short-answer and long-answer questions, **please box or circle your final answer** (unless it is an explanation).

Grading Sheet (for instructor use only)

Question	Points	Score
1	53	
2	22	
3	11	
4	14	
Total:	100	

Name: _____

Honor Code (adapted from Dr. Elaine Rich)

The University and the Department are committed to preserving the reputation of your degree. In order to guarantee that every degree means what it says it means, we must enforce a strict policy that guarantees that the work that you turn in is your own and that the grades you receive measure your personal achievements in your classes:

By turning in this exam with your name on it, you are certifying that this is yours and yours alone. You are responsible for complying with this policy in two ways:

1. You must not turn in work that is not yours or work which constitutes any sort of collaborative effort with other students.
2. You must take all reasonable precautions to prevent your work from being stolen. It is important that you do nothing that would enable someone else to turn in work that is not theirs.

The penalty for academic dishonesty will be a course grade of F and a referral of the case to the Dean of Students Office. Further penalties, including suspension or expulsion from the University may be imposed by that office.

Please sign below to indicate that you have read and understood this honor code.

Signature: _____

Part 1: Multiple Choice / Short Answer (53 points)

1. (53 points) Answer these questions by giving the option or options corresponding to the answer (3 points each unless otherwise specified). **If given letter options, give exactly one answer. If given roman numeral options, select all that apply. Carefully read the instructions on each question.**

   You will receive partial credit on "select all that apply" questions for having partially correct answers.

_____ II, III, IV(1; 4 points) Which statements about classification below are true? **Select all that apply.**

I. The perceptron algorithm is always guaranteed to converge.
II. Logistic regression with bag-of-words features is a convex optimization problem
III. Stochastic gradient descent can be used to optimize a logistic regression model
IV. The perceptron algorithm can be interpreted as an instance of stochastic gradient descent with a certain loss function
V. In the limit as epochs go to infinity, logistic regression and perceptron give the same decision boundary
VI. In the limit as epochs go to infinity, logistic regression and perceptron give the same decision boundary if the data are linearly separable

VI is not right because the actual boundary / weight vector is not the same, even if they'll make the same classification decision on the training points

_____ A(2) Suppose we want a system to take as input some text $\mathbf{x}$ (sentence, document, etc.), then output a label y from a set of five class labels $\mathcal{Y}$. What is the most correct name for this type of machine learning problem?

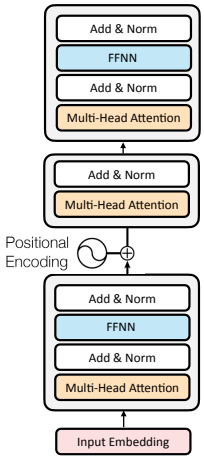
A. Classification
B. Language modeling
C. Clustering
D. Syntactic parsing

_____ D(3) You build a sentiment analysis system that feeds word tokens into a unidirectional RNN, then outputs the sentiment class by putting the final hidden state through a linear + softmax layer. You observe that your model incorrectly predicts very positive sentiment for the following (negative sentiment) passage: *The play was terrible. The performances were lackluster and the acting was unconvincing. Then there was long line to exit the theatre building. At least the dinner was excellent.* Why might the model make this decision?

A. RNNs are not good models for sequence classification tasks
B. RNNs do not model the unknown words like *lackluster* well
C. RNN training is very unstable
D. An RNN's state is heavily influenced by recent tokens

_____ I, II, III(4; 4 points) Considering the example from the previous question: What other methods besides RNNs might work better *for this example*? **Select all that apply.**
I. A bag-of-words model expected to work okay because there are more negative words anyway
II. A deep averaging network

III. A Transformer

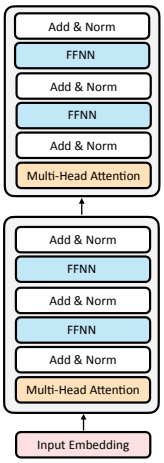


_____ C(5) Here is a picture of a modified 3-layer Transformer (above). How many parameters will this have compared to the standard 3-layer Transformer?

A. More
B. The same
C. Fewer

_____ C(6) How will this do compared to the standard Transformer *in aggregate, across many tasks*?

A. Better
B. The same
C. Worse



_____ C(7) Here is another picture of a modified 2-layer Transformer (above). How many parameters will this have compared to a standard **4-layer** Transformer?

A. More
B. The same
C. Fewer

_____ I, III(8) What types of parameters can be learned in a deep averaging network? **Select all that *could* be learned.**

I. Feedforward layer parameters
II. W^K (weights mapping embeddings to keys for self-attention)
III. Word embeddings
IV. Attention map weights

_____ A(9) Which of the following would be most likely to improve the accuracy of a pre-trained BERT model when fine-tuned on downstream tasks?

A. Having it mask more than 15% of the tokens
B. Having it only do left-to-right (unidirectional) encoding
C. Instead of masking tokens at the input and feeding in [MASK], replacing them with random other words and feeding those words

_____ II, III(10; 4 points) Suppose you are training a language model as in Assignment 3. Which of the following is a possible correct shape for the attention map? **Select all that apply.**

I. [batch size, seq len]
II. [seq len, seq len]
III. [batch size, seq len, seq len]
IV. [batch size, seq len, $\mathrm{d}_{\mathrm{model}}$]
V. [batch size, $\mathrm{d}_{\mathrm{model}}$, $\mathrm{d}_{\mathrm{model}}$]
VI. [batch size, d_k, d_k]
VII. [d_k, d_k]

_____ V(11; 4 points) Suppose you are training a language model as in Assignment 3. Which of the following is a possible correct shape for the model outputs that are input to the loss function? **Select all that apply.**

I. [batch size, seq len]
II. [batch size, seq len, seq len]
III. [batch size, vocab size]
IV. [batch size, seq len, $\mathrm{d}_{\mathrm{model}}$]
V. [batch size, seq len, vocab size]

_____ I, II, III(12; 4 points) Suppose you train a model for a sentence-level classification task (like sentiment analysis) over a dataset containing only singular nouns. You then apply it to a test set also containing plural nouns. Which of the following *could* help your model generalize better? Select all that apply.

I. Stemming
II. Subword tokenization
III. Using pre-trained word embeddings
IV. Using a bag-of-words featurization answer doesn't make sense
V. Running a syntactic parser

Suppose you have the following probabilities from a language model:

$P(\text{a}) = 0.8$
$P(\text{the}) = 0.15$
$P(\text{dog}) = 0.05$

$P(\text{a} \mid \text{a}) = 0.01$	$P(\text{a} \mid \text{the}) = 0.15$	$P(\text{a} \mid \text{dog}) = 0.3$
$P(\text{the} \mid \text{a}) = 0.01$	$P(\text{the} \mid \text{the}) = 0.01$	$P(\text{the} \mid \text{dog}) = 0.5$
$P(\text{dog} \mid \text{a}) = 0.98$	$P(\text{dog} \mid \text{the}) = 0.84$	$P(\text{dog} \mid \text{dog}) = 0.2$

Assume that generation always terminates after two tokens are generated.

_____ 9(13) How many possible sequences could be returned by *sampling*? Give your answer as an integer.

_____ 3(14) How many possible sequences could be returned by *nucleus sampling* with $p = 0.9$? Give your answer as an integer. Recall that nucleus sampling truncates the distribution as follows: take the top probability options from the distribution at a given step until that probability exceeds p, then ignore the rest of the options.

two choices to start, then two choices from the but only one choice from a

The following questions deal with the Viterbi algorithm, reproduced below.

Algorithm 1 Viterbi Algorithm

```
1:  function VITERBI(x, S, T, E)   ▷ x: sentence of length n, S: initial log probs, T: transition log probs,
    E: emission log probs. U denotes the tagset
2:      Initialize v, a n × |U| − 1 matrix
3:      for y = 1 to |U| − 1 do                                   ▷ Handle the initial state
4:          v[1, y] = S[y] + E[y, x_1]
5:      end for
6:      for i = 2 to n do
7:          for y = 1 to |U| − 1 do
8:              v[i, y] = E[y, x_i] + max_{y_prev} (T[y_prev, y] + v[i − 1, y_prev])
9:          end for
10:     end for
11:     Handle step and reconstruct sequences (omitted)
12: end function
```

_____ $O(nk^2)$(15) In an example with n words and k tags in the tag vocabulary, what is the runtime of Viterbi? Give your answer in big-O notation. (E.g., $O(n)$ means that the time is linear in the number of words.)

_____ $O(k^n)$(16) In an example with n words and k tags in the tag vocabulary, how many total paths through the sentence (e.g., possible tag sequences) does Viterbi search over? Give your answer in big-O notation.

Part 2: Long Answer (47 points)

2. (22 points) Suppose you have the bag-of-words vocabulary [good, great, not]. Your training examples are pretty simple but have some typos in them:

   $\mathbf{x}$ =goodx $y = +$
   $\mathbf{x}$ =great $y = +$
   $\mathbf{x}$ =not greatq $y = -$
   $\mathbf{x}$ =not good $y = -$

   a (4 points). Write down the feature vector for each of these examples. Pay close attention to the bag-of-words vocabulary above and use that ordering when listing your feature vectors.

   [000]
   [010]
   [001]
   [101]

   b. (6 points) Run perceptron for one epoch on this data, in order. Initialize with $\mathbf{w} = \mathbf{0}$. Use the decision rule $\mathbf{w}^\top f(\mathbf{x}) \geq 0$, where a score of 0 is classified as positive. Report the final weight vector at the end of that epoch.

   Only update is on the third example, gives [0 0 -1] as weights

c. (5 points) Suppose that you are using subword tokenization with the **subword vocabulary** {*good, x, q, not, gre, at, great*}. use this vocabulary and standard (greedy) tokenization to segment each word. Then, give (a) a new bag-of-words vocabulary (replacing the one at the start of the question); (b) features for each examples in your vocabulary.

goodx: good x
(1,1,0,0,0,0,0)
great: great
(0,0,0,0,0,0,1)
not greatq: not great q
(0,0,1,1,0,0,1)
not good: not good
(1,0,0,1,0,0,0)

d. (3 points) Now assume that you had the word *greqat* (with the typo of *q* in the middle of the word). What segmentation will this receive?

gre q at

e. (4 points) For this problem, where you might see typos anywhere in the word, which do you think is more effective at improving accuracy and robustness to typos, independent of runtime: (1) subword tokenization as described above, or (2) explicitly repairing typos by finding the word in the vocabulary with lowest edit distance to the given word? Give a one-sentence justification of your choice.

Typo repair will generally be less fragile. Subword tokenization in this case breaks up the words in weird ways. It might work okay in general, but for the two examples shown here, typo repair will fix them whereas subword tokenization breaks up gre q at.

3. (11 points) Suppose you have word embeddings for three words 1, 2, and 3.

$v_1 = (0, 0)$
$v_2 = (1, 1)$
$v_3 = (-1, 1)$

$c_1 = (0, 0)$
$c_2 = (0, 1)$
$c_3 = (1, 0)$

Assume for this question that e (the mathematical constant) is equal to 3.

a. (4 points) What is the distribution over context words for v_1?

1/3 1/3 1/3

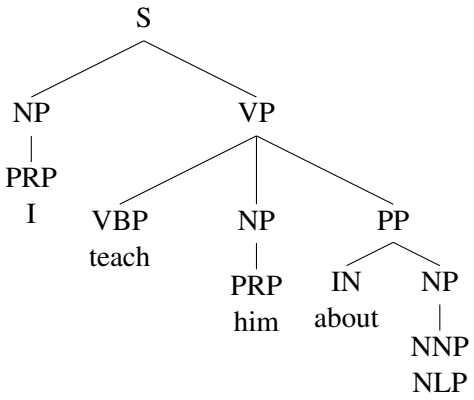
b. (4 points) What is the distribution over context words for v_2?

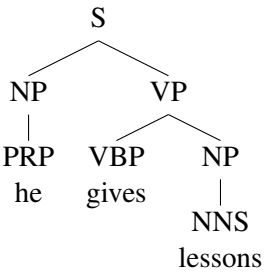
1/7 3/7 3/7

c. (3 points) Suppose you have two other words 4 and 5. These two words cooccur with each other: 4 occurs with 4, 4 with 5, and 5 with 5. Neither cooccurs with the existing words. What is the *minimum* number of extra dimensions you need to represent these probabilities while preserving the existing relations? Briefly justify your answer, but you do not need to construct actual vectors.

The intended answer was 1. However, due to a late modification in this question, the zero vectors for v and c make it impossible to achieve good "saturation" of the probabilities. If we set that aside, there was an unintended solution using -10 in the second position that achieves good probabilities without any extra dimensions. Therefore, we awarded fully credit for any answer here.

4. (14 points) Consider the following trees:



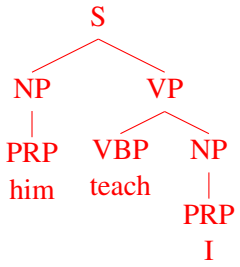


a. (6 points) Write down the rules for a grammar extracted from these trees, respecting the following:

1. First list rules rooted in S, then those rooted in NP, then VP, then PP (hint: you will have rules rooted in each of these symbols)
2. You do NOT need to report probabilities.
3. You do NOT need to include the lexicon (any rule from a tag to a word), only "internal" rules starting in nonterminal categories listed above
4. Finally, do not do binarization or any other kind of preprocessing.

S → NP VP
NP → PRP
NP → NNS
NP → NNP
VP → VBP NP PP
VP → VBP NP
PP → IN NP

b. (5 points) Draw the parse tree for the sentence *him teach I*, or write *not parseable* if no tree can be produced. Hint: you should not need to formally run CKY. Try to see what symbols can be built over each span of this sentence.



c. (3 points) The result on the previous part is not completely satisfactory from a linguistic perspective. In one sentence, describe one thing you could do to modify your grammar in order to make it produce more linguistically "correct" behavior on the example.

Applying vertical Markovization to make NP^S and NP^VP symbols is the "best" answer. There are other choices too, like other kinds of grammar refinements

Extra credit (2 points): Describe in one sentence how the word embedding debiasing method of Bolukbasi et al. (2016) works. (It's the debiasing technique we discussed in class.)

It "neutralizes" words with respect to a vector subspace defined by gendered features (or bias information in general).