Midterm for CS388: Natural Language Processing (Fall 2023)

Instructions:

- You will have 150 minutes (2.5 hours) to complete and upload the exam once you start it.
- **The first thing you see on Gradescope will be the Honor Code, which you will read and (electronically) sign.** This exam is to be completed individually be each student. Again, **you may not collaborate with other students**! If we find out that you have done so, that will be considered a violation of the course Academic Honesty policy.
- This exam is an **open book take-home exam**. You are allowed to consult any *written* resources that are helpful. **You may not collaborate with any other people or AI agents.**
- **You may not use ChatGPT.**
- Partial credit will be given for short-answer and long-answer questions, so please show work in your answers, but avoid writing essays. **You might be penalized for writing too much if it's incorrect.**
- For long-answer questions, **please box or circle your final answer** unless it is an explanation.
- Multiple-choice and short-answer questions will be entered directly into Gradescope. Long-answer questions should be uploaded as PDFs. You may type your responses to these questions, handwrite responses on a printed exam, or anything else.
- Because of the asynchronous nature of the exam, **we cannot clarify anything in the exam.**

Grading Sheet (for instructor use only)

Question	Points	Score
1	52	
2	18	
3	13	
4	17	
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 (52 points)

1. (52 points) Answer these questions by giving the option or options corresponding to the answer (2 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

_____ I, III(3) 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

_____ C(4) Which of the following is true of the debiasing method of Bolukbasi et al. (2016)?

A. Debiasing reduces the vocabulary by eliminating potentially biased words
B. Debiasing shrinks the dimensionality of word embeddings to remove bias
C. Debiasing "neutralizes" words with respect to a gender subspace
D. After debiasing, we do not expect a word like "homemaker" to have high similarity with any historically gendered concepts

_____ D(5) 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(6; 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

Consider the following table of bigram counts, where the count (a, b) means the number of times b was observed following word a:

(the, frog) 4
(frog, is) 2
(frog, was) 4
(is, <UNK>) 1
(is, frog) 1
(<UNK>, <UNK>) 0

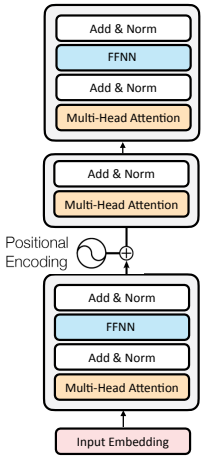
Assume that these are the only words observed in the model. Assume that the model's vocabulary consists of {*the*, *frog*, *is*, *was*} and all other words are mapped to UNK. Assume that P(the | <s>) = 1

_____ D (7) What is the probability of "*the frog is blue*" under a bigram language model with no smoothing estimated based on these counts?

A. 1/2
B. 0
C. 1/4
D. 1/6 1 * 1 * 1/3 * 1/2

_____ A (8) Which of the following sentences, if added to the corpus, would increase the probability of *the frog is blue* by the largest amount? Continue to assume that P(the | <s>) = 1, but all other bigram probabilities will be re-estimated with the added sentence.

A. the frog is black Changes the probs to 1 * 1 * 3/7 * 2/3 = 2/7
B. frog is frog is frog is
C. the is blue
D. frog frog frog frog

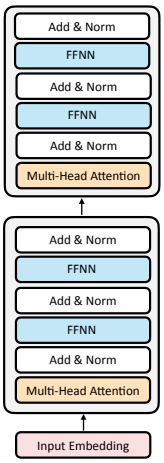


_____ C(9) Here is a picture of a modified 3-layer Transformer (above). How many parameters will this have compared to the standard 3-layer Transformer?

A. More
B. The same
C. Fewer

_____ C(10) How will this do compared to the standard Transformer *in aggregate, across many tasks*?

A. Better
B. The same
C. Worse



_____ C(11) 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

_____ A(12) 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. Having 100% of masked tokens replaced with a random word, instead of standard masking

_____ II, III(13; 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, d_{model}]
V. [batch size, d_{model}, d_{model}]
VI. [batch size, d_k, d_k]
VII. [d_k, d_k]

_____ V(14; 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, d_{model}]
V. [batch size, seq len, vocab size]

_____ I, II, III(15; 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/help
V. Running a syntactic parser not clear how this would help without more detail

---

Suppose you have the following probabilities:

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

$P(\text{a} \mid \text{a}) = 0.01$
$P(\text{the} \mid \text{a}) = 0.01$
$P(\text{dog} \mid \text{a}) = 0.98$

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

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

Assume that generation always terminates after two tokens are generated.

_____ a dog, the dog(16) What sequences are returned by beam search with beam size = 2? Give the sequences as a comma-separated list of strings.

_____ 9(17) How many possible sequences could be returned by sampling? Enter your answer as an integer.

_____ 3(18) How many possible sequences could be returned by nucleus sampling with $p = 0.9$? Enter your answer as an integer. two choices to start, then two choices from the but only one choice from a

---

The following questions deal with the Viterbi algorithm.

_____ $O(nk^2)$(19) In an example with n words and k tags per timestep, 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.)

_____ $O(k^n)$(20) In an example with n words and k tags per timestep, how many paths does Viterbi search over? Give your answer in big-O notation.

_____ $O(kl^{n-1})$(21) Now suppose that each tag can only be followed by at most l other tags. How many paths are there? Give your answer in big-O notation using the constants n, k, and l.

Part 2: Long Answer (48 points)

2. (18 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 (3 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. Write this as a comma-separated list of vectors like $[x_{11}, x_{12}, ...], [x_{21}, x_{22}, ...], ...$

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

   b. (5 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 in the format $[w_1, w_2, ...]$.

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

   c. (5 points; freeform answer) 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)

   The question was meant to say "feature for each example". We awarded partial credit if features were computed for the vocabulary instead of the examples.

d. (2 points; freeform answer) 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. (3 points; freeform answer) 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. (13 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. Also assume that e^{10} is effectively "infinity" from the standpoint of softmax computations, and e^{-10} is effectively "-infinity".

a. (3 points) What is the distribution over context words for v_1? Write your answer as a tuple of fractions like (3/4, 1/4) (except in your case you should report three values, for words 1, 2, and 3 in order).

1/3 1/3 1/3

b. (3 points) What is the distribution over context words for v_2? Report your answer following the same convention as in part (a).

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?

1 was the intended answer; see part (d) below for more discussion

d. (4 points; freeform answer) Assume that the existing word and context vectors for words 1, 2, and 3 are extended with 0s in the new dimensions you needed from part (c). Now give a set of word and context vectors for words 4 and 5 (*only* providing word and context vectors for these examples) with the following properties:

1. These vectors appropriately model the relationship between words 4 and 5
2. These vectors correctly model the relationship between words 4 and 5 and the original words 1-3

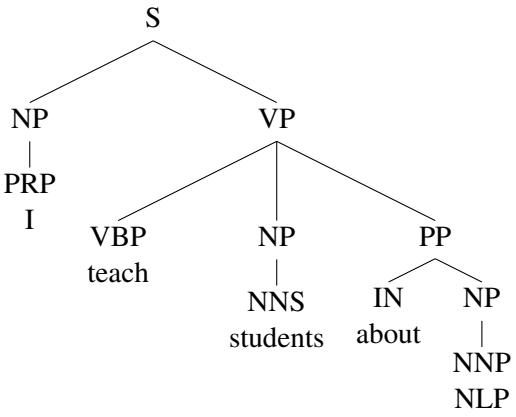
You may use 10 as a sufficiently large value for logits (assume e^{10} is arbitrarily large and e^{-10} is arbitrarily small, as before).

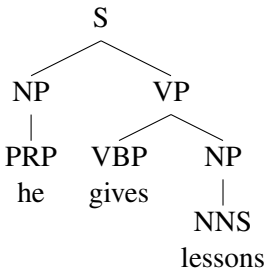
This question had a bug in that the zero vector for v_1 ended up making the intended "saturation" impossible for the v_1 vector, which will always place a uniform distribution over the contexts. Because of this, you could actually get a nearly optimal solution with 0 extra dimensions with the following vectors:

$v_4 = (0, -10)$
$v_5 = (0, -10)$
$c_4 = (0, -10)$
$c_5 = (0, -10)$

The original intended solution of the solution involved expanding by 1 dimension with a similar vector structure

4. (17 points) Consider the following trees:



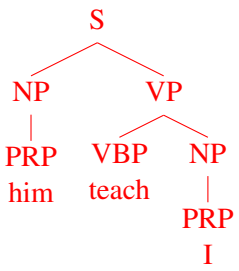


a. (5 points; freeform answer) Write down the rules for a grammar extracted from these trees, respecting the following:

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. (4 points; freeform answer) 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.



Due to a last-minute modification of the question (there were initially more/more complex trees), *him* ended up being removed from the lexicon. We also accepted not parseable, although the intention was to illustrate that the accusative "him" can show up as a subject in this grammar.

c. (3 points; freeform answer) Now suppose that we do a single step of **vertical Markovization**, but only to the NP symbols. That is, we replace each NP with a new symbol NPˆX where X is the parent node of that NP. For example, the NP above *I* becomes NPˆS. **This operation is applied to the tree before the grammar is extracted.**

Write down the new grammar extracted from the tree, following the same rules as for part (a). List all rules rooted in the new NP symbols together.

S $\rightarrow$ NPˆS VP
NPˆS $\rightarrow$ PRP
NPˆVP $\rightarrow$ NNS
NPˆPP $\rightarrow$ NNP
VP $\rightarrow$ VBP NPˆVP PP
VP $\rightarrow$ VBP NPˆVP
PP $\rightarrow$ IN NPˆPP

d. (3 points; freeform answer) With the new grammar, draw the parse tree for the sentence *him teach I*,

or write *not parseable* if no tree can be produced.

not parseable. PRP cannot occur under NPˆVP

e. (2 points; freeform answer) How does the set of sentences defined under the grammar with vertical Markovization compare to the set of sentences from the grammar without vertical Markovization? Is it a superset, subset, the same set, or neither (a different but unrelated set)? You do not need to give an explanation.

It is a subset, because the NPs can no longer be substituted between being under S and under VP.