Practice Exam for CS505: Natural Language Processing (Spring 2026)

Instructions

- You will have 75 minutes (our usual class time) to complete the exam. We recommend spending 15–20 minutes total on multiple choice, 20–25 minutes total on short answer, and 20–30 minutes total on the multi-part short answer sections. Use any remaining time to double-check your answers, or to revisit questions you weren't sure about.
- This practice exam shows you the general format of the exam. It does **not** reflect the exact questions that will be on the exam, but the topic of the questions will be similar.
- The exam must be completed individually by each student.
- You are allowed one 8.5" $\times$ 11" double-sided note sheet. You must turn in your note sheet alongside your exam.
- You are **not** allowed to use any electronic tools, such as phones, laptops, or calculators.
- For short-answer and multi-part questions, partial credit will be given for showing correct work, even if you arrive at the incorrect final answer.
- Please write your name on each page.

Name: ____

Multiple Choice (41 points)

The following questions are worth 3 points each, unless otherwise indicated. When given letter options, give exactly one answer. When given roman numeral options, select all that apply. Circle your answers, and optionally write the letter(s) you chose in the line next to the question numbers. You will receive partial credit on "select all that apply" questions for having partially correct answers.

B. 1. Say you have an input text sequence $\mathbf{x}$, and your goal is to select a label for the sequence from a set of 3 labels. Which of the following is the most accurate description of this task?

- A. Language modeling
- B. Classification
- C. Parsing
- D. Sequence labeling

II. 2. (4 points) You train a text classifier on a dataset where 90% of the examples are class A, and the remaining 10% are class B. Your model achieves an accuracy of 0.90 (on a scale from 0 to 1) on the training data, but when you evaluate on class B examples only, it correctly classifies 10% of them. Which of the following are likely to be true? Select all that apply.

- I. The recall is high for class B.
- II. The model may be predicting class A for nearly all examples.
- III. The macro-F_1 score will also be approximately 0.90.
- IV. On a balanced test set, the model will likely get around 0.90 accuracy.

C. 3. You have two language models. Model A uses a character-level vocab of size 100, and Model B uses a word-level vocab of size 50000. Model A gets a perplexity of 5, and model B gets a perplexity of 50. Which of these is true?

- A. Model A is better because it obtains a lower perplexity.
- B. Model B is better because it obtains a higher perplexity.
- C. It's not clear which model is better because we can't compare the perplexities of models with different vocab sizes.
- D. It's not clear which model is better because the perplexities are too close.

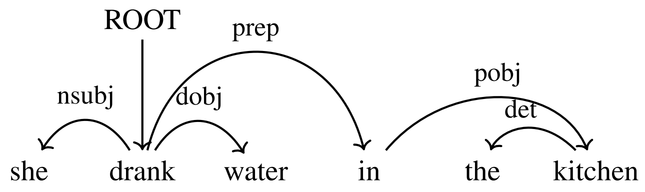
A. 4. In decoder-only Transformers, we have a causal mask. What does the causal mask do?

- A. Prevents the model from looking at future tokens when processing a token.
- B. Prevents the model from looking at future tokens *and* the current token when processing a token.
- C. Allows the model to look at all other tokens (before and after the current token) when processing a token.
- D. Only allows the model to look at the current token being processed.

D. 5. A bigram classifier is more expressive than a unigram classifier, yet it can sometimes perform worse when we have limited data. Which of the following best explains this?

- A. Bigram features capture less meaning than unigrams.
- B. Bigram classifiers can't be trained with logistic regression.
- C. The softmax function is undefined for high-dimensional feature vectors.
- D. Bigrams lead to much sparser features than unigrams, which leads to overfitting.

I.,III.,IV. 6. (4 points) Consider the following dependency parse:



Which of the following are true about this parse? Select all that apply.

I. This dependency graph is projective.

II. "drank" has one dependent.

III. "in" is the head of the *pobj* dependency.

IV. The fact that "in" is attached to "drank" and not "water" implies that "in the kitchen" describes where she drank the water, rather than where the water was.

II.,IV. 7. (4 points) Which of the following are true statements about constituencies and dependencies? Select all that apply.

I. Dependencies can represent recursive structures. Constituencies cannot.

II. Constituency trees assume that words can be grouped into larger structures, whereas dependencies only assume relationships between pairs of words.

III. Dependencies are a model of syntax, while constituencies are not.

IV. In constituency trees, every word is a leaf node. In dependency trees, the same word can potentially be a head of one word *and* a dependent of another.

B. 8. Recall that "NN" is a POS tag referring to singular common nouns, and "VBS" is a POS tag referring to 3rd-person singular verbs. In a Hidden Markov model POS tagger, the transition probability $p(\text{VBS}|\text{NN})$ represents:

A. The probability that the current word is a verb.

B. Given the previous tag was a noun, the probability that the current tag is a verb.

C. Given the current tag is a verb, the probability the next tag will be a noun.

D. The joint probability of seeing a noun and verb together.

A. 9. You have a bigram language model whose vocabulary contains 5 types. If you generate 3 tokens, how many strings are possible if you use *greedy decoding*?

A. 1

B. 3

C. 5

D. 125

D. 10. Like in the previous question, you have a bigram language model whose vocabulary contains 5 types. If you generate 3 tokens, how many strings are possible if you use *(pure) sampling*? Assume that each token in the vocabulary always has a non-zero probability according to the model.

A. 3

B. 5

C. 25

D. 125

D. 11. You are running byte-pair encoding (BPE) on this text sequence that has been split into characters: "t h e _ t h i n g _ t h e r e". What would be the first and second new tokens added by BPE?

A. (i) th, (ii) he

B. (i) ng, (ii) ing

C. (i) he, (ii) her

D. (i) th, (ii) the

I.,III.,IV. 12. (5 points) Which of the following are true of Transformers and recurrent neural networks (RNNs)? Select all that apply.

I. During training, RNNs must process tokens in a document sequentially, whereas Transformers can process multiple tokens in parallel.

II. Transformers do *not* need positional information in their embeddings because self-attention is inherently position-aware.

III. LSTMs mitigate vanishing gradients compared to regular RNNs using their gating mechanisms.

IV. If you remove residual connections and LayerNorms, Transformers become more susceptible to vanishing or exploding gradients.

V. RNNs cannot model dependencies between tokens without self-attention.

Short Answer (24 points)

The following questions are each worth 6 points. We recommend spending no longer than 5 minutes on each question. If you get stuck, move on and revisit this at the end.

______ 1. (6 points) Here is a table containing the true labels for a set of sentiment classification examples, compared to a classifier's predictions for those examples.

Example	**True**	**Predicted**
1	$+$	$+$
2	$-$	$-$
3	$-$	$+$
4	$+$	$-$
5	$-$	$-$
6	$+$	$+$
7	$-$	$+$
8	$-$	$-$

For the **positive class** ($+$), compute the precision (2 points), recall (2 points), and F_1 score (2 points) of the classifier. You do not have to simplify; you can leave your answers as fractions, or products/divisions of fractions. Note that you are *not* computing the macro-F_1, just the F_1 for the positive class. Show your work so that we can assign partial credit if needed.

Precision: $\frac{2}{4} = \frac{1}{2}$

Recall: $\frac{2}{3}$

F_1: $\frac{2 \times P \times R}{P+R} = \frac{2 \times \frac{1}{2} \times \frac{2}{3}}{\frac{1}{2}+\frac{2}{3}} = \frac{\frac{2}{3}}{\frac{7}{6}} = \frac{12}{21} = \frac{4}{7}$. Any of the last four terms would be accepted.

______ 2. (6 points) Take the following dataset, where each document has its own line:

- strawberry is red
- orange is orange
- pear is green

You're training a skip-gram model with a context window size of ± 1, and with $k = 3$ negative samples per positive sample. For the positive example ($w =$ strawberry, $c_+ =$ is), list 3 valid negative samples. In your samples, you should explicitly mark which token is w and which is c_-; otherwise, we will assume w is first and c_- is second.

Any 3 of the following are valid: ($w =$ strawberry, $c_- =$ strawberry), ($w =$ strawberry, $c_- =$ red), ($w =$ strawberry, $c_- =$ orange), ($w =$ strawberry, $c_- =$ pear), ($w =$ strawberry, $c_- =$ green).

______ 3. (6 points) You've trained an encoder-only Transformer. Here are the queries q and keys k it computed for each token in the sentence "cats sleep quietly":

Word	q	k
cats	[0, 1, 2]	[2, 0, 0]
sleep	[1, 2, 0]	[0, 2, 0]
quietly	[0, 2, 0]	[0, 0, 2]

Your model is currently processing the token "cats". Which token in the sequence will have the highest attention score with "cats"? You only have to provide the correct token, but we encourage you to explain your reasoning so we can give partial credit if needed.

"quietly". We're currently processing "cats", so we're using the query vector for "cats". The key vector with the highest similarity to the "cats" query vector corresponds to "quietly".

______ 4. (6 points) You have a bigram language model with the following probabilities:

```
p(x) = 0.6     p(y) = 0.2     p(z) = 0.1

p(x|x) = 0.2   p(y|x) = 0.6   p(z|x) = 0.2
p(x|y) = 0.5   p(y|y) = 0.1   p(z|y) = 0.1
p(x|z) = 0.8   p(y|z) = 0.1   p(z|z) = 0.1
```

What is the perplexity of this model on the test sequence "`z x y`"? You do not have to simplify; you can leave your answer in terms of e and $\ln$ (or in terms of probability multiplications and exponents/roots if you choose not to use log-probabilities).

You have a few possible right answers:

$\exp(-\frac{1}{3}(\ln 0.1 + \ln 0.8 + \ln 0.6))$

OR

$\sqrt[3]{\frac{1}{0.1\times0.8\times0.6}}$

OR

$(0.1 \times 0.8 \times 0.6)^{-\frac{1}{3}}$

Multi-part Short Answer (35 points)

We recommend spending 10–15 minutes on each question in this section (or about 2–5 minutes per subpart, depending on how difficult they are). If you get stuck, move on and come back later.

_____ 1. We have the following corpus, where each document has its own bullet. Co-occurrences do *not* cross bullet boundaries.

- <s> a c c b </s>
- <s> a b c a </s>
- <s> b c c a </s>

(a) (4 points) Assume we're training a bigram language model with vocabulary $V = \{$a, b, c, <s>, </s>$\}$. If we do *not* use smoothing, what is $p(c|a)$? You may leave your answer as a fraction.

$\frac{C(a,c)}{C(a)} = \frac{1}{4}$

(b) (6 points) The bigram "b a" never appears in the corpus. If we do not use smoothing, what probability does the bigram model assign to the sequence "b a c" (3 points)? In one sentence, explain why this is a problem (3 points).

The probability will be 0. This is a problem because our model will assign *any* sequence containing "b a" a probability of 0, making our model useless on many examples.

(c) (4 points) Compute the probability of $p(a|b)$ with Laplace smoothing $(k = 1)$. You may leave your answer as a fraction.

$\frac{C(b,a)+k}{C(b)+V} = \frac{0+1}{3+5} = \frac{1}{8}$

(d) (3 points) A classmate trains an RNN on the data and achieves a lower perplexity on held-out data compared to your bigram language model with smoothing. In one sentence, give one reason why an RNN might outperform a bigram model.

RNNs generalize more effectively because they embed tokens into meaningful representations, instead of just storing token co-occurrence counts.

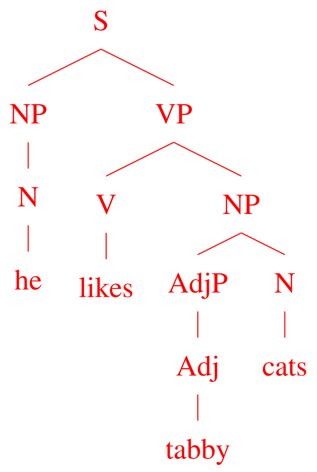
OR

RNNs can model longer contexts than bigrams using their context vector.

______ 2. You have the following probabilistic context-free grammar (PCFG), with probabilities given in brackets next to each rule.

```
[1.0] S -> NP VP      [0.25] N -> he
[0.5] NP -> N         [0.25] N -> him
[0.5] NP -> AdjP N    [0.25] N -> cat
[0.5] VP -> V         [0.25] N -> cats
[0.5] VP -> V NP      [1.0]  Adj -> tabby
[1.0] AdjP -> Adj     [0.5] V -> like
                      [0.5] V -> likes
```

(a) (6 points) Draw the constituency tree for the sentence "he likes tabby cats".



(b) (4 points) Given your tree from part (a), provide the probability of the NP containing "tabby cats". You do not have to simplify; you can leave your answer as a product of probability numbers.

$p(\text{NP} \rightarrow \text{AdjP N}) \times p(\text{AdjP} \rightarrow \text{Adj}) \times p(\text{Adj} \rightarrow \text{tabby}) \times p(\text{N} \rightarrow \text{cats}) = 0.5 \times 1.0 \times 1.0 \times 0.25$

(c) (5 points) Add some rules to the grammar that would allow your grammar to parse the sentence "he likes cats with spots". Here are a few to get you started: "P $\rightarrow$ with", "N $\rightarrow$ spots". (You do not need to provide probabilities, and you do not need to modify the probabilities of existing rules to accomodate these new ones.) You should only need **two** or **three** additional rules.

NP $\rightarrow$ N PP

PP $\rightarrow$ P NP

(d) (3 points) Our grammar can parse sentences like “him likes cat”. It is not ideal that our grammar can parse sentences that we wouldn’t find grammatical. In 1–2 sentences, describe what linguistic information is missing from the grammar that allows these ungrammatical sentences to be generated.

Our grammar does not capture the distinction between subjects and objects, which is why “him” can be a subject.