CMSI 537
Take-home Midterm

Name:
Due beginning of class Tuesday, 10/27.

Instructions

This take-home exam is open book and open notes. However, please do not consult or work with anyone else on any of these problems, and do not look them up on the Internet. You are welcome to run your code as you test and debug it, and you may take as much time as you want to work on it. You can ask me clarification questions, but not how to solve it. If a template file is not provided for you, create a new file yourself.

Honor Code Pledge

Write out the pledge quoted here, and sign your name below. "I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own."

Write the pledge here:

Signature: ____

Bias
What is one example of bias in NLP, and how would you approach addressing it?

Theory: Complete **TWO** of the three theory questions below. Answering the third question is optional.

0. **Language Models**

   Assume you are given the following corpus of text:

   $\langle s \rangle$ *I love NLP* $\langle e \rangle$
   $\langle s \rangle$ *You love programming* $\langle e \rangle$

   (a) Provide all the bigram probabilities for this corpus.

   (b) Calculate the probability for the sentence $\langle s \rangle$ *You love NLP* $\langle e \rangle$.

   (c) What is the perplexity of the corpus? (Note: Since we are using bigrams here, you can assume $P(\langle s \rangle) = 1$, since it always starts the sentence).

   (d) What is the probability of the sentence $\langle s \rangle$ *I like NLP* $\langle e \rangle$? What is the probability if we adjust all the bigrams with add-one (Laplace) smoothing? Be sure to show your work by writing out all the new bigram probabilities that you need.

1. **Backpropagation**
Consider the following 2-layer, feed-forward (FF) neural network:

$$\boldsymbol{h}_1 = \tanh(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) \tag{1}$$

$$\boldsymbol{h}_2 = \tanh(\boldsymbol{W}_2\boldsymbol{h}_1 + \boldsymbol{b}_2) \tag{2}$$

$$\hat{y} = \sigma(\boldsymbol{w} \cdot \boldsymbol{h}_2 + b) \tag{3}$$

Write the gradients with respect to the loss $L = -\log(\hat{y})$ for each parameter (i.e., weights $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{w}$ and bias terms $\boldsymbol{b}_1, \boldsymbol{b}_2, b$.

2. **A Probabilistic Context-Free Grammar (PCFG)**
Consider the following PCFG, where the top half of the table is the grammar, and below is the lexicon:

Rule	*P*(*Rule*)
S → NP VP	1.0
NP → DT NN	0.9
NP → NP PP	0.1
VP → VP PP	0.2
VP → VP NP	0.5
PP → IN NP	1.0
IN → *with*	1.0
DT → *the*	1.0
VP → *saw*	0.3
NN → *woman*	0.4
NN → $professor$	0.4
NN → *telescope*	0.2

(a) For the sentence *the woman saw the professor with the telescope*, how many valid parses are there? Draw the parse tree(s).

(b) What is/are the probability(ies)? If more than one, which is largest and what is the meaning or interpretation of that parse tree?

Programming
Go to this link to create your own private repository and to import the data I have provided for you: https://classroom.github.com/a/xhw5F3c4

0. **Semantic Tagging**—`tagger.py`
For this problem, you will be working with a real-world dataset LMU student Maya Epps collected on the crowdsourcing platform Amazon Mechanical Turk for a research project on diet and exercise logging with natural spoken language. Your goal is to tag every sentence in the provided CSV data file as one of the following: BE (Begin-Exercise), IE (Inside-Exercise), BF (Begin-Feeling), IF (Inside-Feeling), or O (Other), as illustrated in the diagram below.

Exercise	I	just	ran	three	miles	on	the	track
Tag:	O	O	BE	O	O	O	O	O

How they felt	I	'm	really	out	of	shape
Tag:	O	O	O	BF	IF	IF

(a) As the first step, you should load the training data and split it into 80% training and 20% validation. The data has already been tokenized, and contains part-of-speech (POS) tags.

(b) As a baseline, build one or two classifiers: Naive Bayes and/or logistic regression. What features did you use? Report precision, recall, and F1 scores per tag on the held-out test data.

(c) Finally, implement a sequence-labeling neural network. As you explore various models and hyperparameters, you should evaluate on the validation data. What architecture and hyperparameters did you choose? How many epochs did you train for? Only after you're all done fine-tuning, evaluate on the held-out test set. How do the results compare to the baseline classifiers?

What to turn in
Submit the files you modified to your GitHub Classroom repository—make sure the code is beautiful, with well-chosen names, perfect formatting, and appropriate comments (if called for).

Upload to Brightspace the pdf with your hand-written pledge and signature, along with all your answers to the written questions, by 10/27.

0. Number of hours spent working on this exam:

1. (Optional) Feel free to let me know what you liked/disliked about this midterm, what you learned, etc: