

SI 630 Midterm Exam

Winter 2022

Instructions: You have 24 hours to finish this exam, though you should need less. The exam consists of two parts: a short answer section worth 20% and a long-answer part worth 80%. There are no late submissions for this exam; exams turned in after the deadline will not be graded.

Important note: Email clarification questions to jurgens@umich.edu; all clarifying answers will be posted to Piazza in a single thread.

Academic Integrity Policy

All submitted work must be your own, original work. **No portion of this exam should be discussed with any other person.** If caught, you will receive a zero for that portion of the exam (or the whole exam), depending on the scope of the plagiarism. All cases will be referred to the Office of Academic Integrity, regardless of their scope. Any violation of the University's policies on Academic and Professional Integrity may result in serious penalties, which might range from failing an assignment, to failing a course, to being expelled from the program.

Warm up question:

By signing your name above, you agree to abide by the honor code above.

Your answer: Chongdan Pan

Uniquename of your answer: pandapcd

Suggestions for this exam:

- Read the entire exam first before starting any questions
 - One option is to read it in the morning, think briefly, then go about your day while you've had time to reflect on the short answer
- Decide to answer the easiest questions first
- Time-box your exam, so that you don't spend too much time working on it
- Remember that short answer is only 20% of this exam
- Use the slides or SLP3 book to look things up when you feel stuck
- After writing your response to a long-answer question, ask yourself what a classmate would ask you about your solution, then include those details, then ask what the

instructors would ask you, then add *those* details, then think what an interviewer asking the question would want to hear and add those details.

- Take deep breath
- Feel free to work in spurts; answer a question, then do something else
- Go for a walk and let the activity jog your brain too—seriously, walks are magic sometimes
- Don't wait until 10pm on the second day to start (please!)

Short Answer (20 points total):

Question 1: Semantics (5 points)

Consider the following six senses for the lemma “test” as a noun, some of which also have example glosses.

1. trial, trial run, **test**, tryout: trying something to find out about it
 - "a sample for ten days free trial"; "a trial of progesterone failed to relieve the pain"
2. **test**, mental test, mental testing, psychometric test: any standardized procedure for measuring sensitivity or memory or intelligence or aptitude or personality etc.
 - "the test was standardized on a large sample of students"
3. examination, exam, **test**: a set of questions or exercises evaluating skill or knowledge
 - "when the test was stolen the professor had to make a new set of questions"
4. **test**, trial: the act of undergoing testing
 - "he survived the great test of battle"; "candidates must compete in a trial of skill"
5. **test**, trial, run: the act of testing something
 - "in the experimental trials the amount of carbon was measured separately"; "he called each flip of the coin a new trial"
6. **test**: a hard outer covering as of some amoebas and sea urchins

Part 1 (4 points): Using the Lesk Algorithm, identify the sense of “test” in the following sentence and show your work for how you decided which sense was present:

- “The students hadn’t failed the test because they studied a little bit in the days leading up to it and completed all the questions.”

Part 2 (1 point): Pick one of the senses and suggest how you would modify the definition or glosses to improve performance on the Lesk algorithm in future cases.

Question 2: Evaluation (5 points)

We’ll use the table below to calculate evaluation metrics in two ways for a sentiment analysis system. The system is a three-class sentiment analysis system with positive, neutral, and negative classes. In this evaluating setting, the system has provided predictions for some

instances that weren't labeled in the answer key and has not provided predictions for some labeled instances in the answer key. If no label is assigned in either the answer key or the system output, it will not be listed below and will not be considered in the score.

Part 1 (2 points): In the first, we'll consider this table as a *three-class* classification problem. Calculate precision, recall and F1 for the following system output and answer key **by hand** (show your work). For the F1, calculate the macro-average (average F1 across all classes).

System Output	Answer Key
negative	neutral
positive	positive
neutral	
negative	positive
positive	positive
	positive
negative	negative
neutral	
negative	neutral
positive	
neutral	negative
neutral	negative
positive	
negative	negative
neutral	positive
	neutral
neutral	
neutral	negative

Part 2 (3 points): Now we'll consider the setup where we *only* care about our model's ability to recognize the negative class. Treat the positive and neural classes as equivalent and compute the *Binary* Precision, Recall, and F1, using "negative" as the positive class in the binary setup. Show your work for making these calculations.

Question 3: Sequence Tagging (5 points)

Consider the following corpus of part-of-speech tagged words, where each line is a separate labeled instance:

- our/J cats/N
- cats/N meow/V
- my/J pet/N
- pet/V the/D cats/N


Part 1 (1 point): Using the four examples above, calculate the maximum-likelihood initial state probabilities for a Hidden Markov Model (HMM) and show your work.

Part 2 (2 points): Using the four examples above, calculate the maximum-likelihood transition probabilities for an HMM estimated from this data. You should include transitions to the STOP symbol in your probabilities.

Part 3 (2 points): Using the words and parts of speech in the training data, write an example of a grammatical English sentence that would be assigned zero probability by the HMM. You can assume that no smoothing has been done to the probabilities.

Question 4: Attention (5 points)

Important note: Remember this is a short answer question and we are not looking for a page-long explanation! You do not need to read anything online (other than the slides) to answer this question for full credit.

You get hired as a data scientist for a company focused on developing corporate mobile apps for team communication. Your boss wants you to develop a classifier that helps end-users by recognizing certain types of actionable messages like  meeting suggestion so they can do in-app pop-ups to take action right away. Your boss estimates you need to recognize around 10 different types of things. A single message might contain a few different types of these things, so you'll need to recognize any that are present. However, your classifier needs to be efficient so you can only build one classifier that produces multi-label output (i.e., can predict multiple things) rather than build separate classifiers for each.

Initially, you were excited to develop a fancy deep learning classifier based on things like BERT, but *unfortunately* the mobile platforms you're working on have no GPUs and are seriously limited in what they can compute. However, you remember there's a simple bag-of-embeddings classifier¹ that also uses "attention" with a learned "important word" vector v and when we compute the inner product $v \cdot w_i$ for each word vector (w_i) and then pass that through a

¹ A simple bag-of-embeddings approach just sums the word embeddings for its input to create a single "document embedding" representation and uses that for classification

softmax, we get an attention weighting to weight the embeddings by their importance, which was described and motivated on Slides 71-87 of the Week 9 lecture.

You are able to fit a sufficient number of embeddings onto the phone that this classifier works pretty well and is very fast. However, the classifier is struggling to recognize some of the **different message types**. Your co-worker suggests adding more than one “important word” vector (i.e., more than one v) to help the model learn different kinds of words to pay attention to for different message types.

Your Task: Describe (i) at a high-level how you could implement a model with multiple “important word” embeddings to support multilabel classification (e.g., what do you do with them?), and (iii) how many additional keyword embeddings would be needed (and why; how many would be too many?).

Long Answer Questions

Instructions: Read the prompts below and write roughly a one-page response to each. There are lots of right answers to each so we’re looking for demonstrations that you can apply what you’ve learned in NLP to solving new kinds of problems. You can safely assume that you have access to all the NLP and machine learning libraries we’ve discussed, so there’s no need to go into algorithmic details unless you think it’s important to demonstrate you know the material or answer a particular question. Also, remember these are real-world scenarios with real problems that often have no “correct solution” and instead both in this exam and in the real world, you’ll have to come up with best-effort solutions.



Most good answers are typically ~1 pages single spaced (1” margins), based on how succinctly you explain your ideas. The questions are designed to push you into slightly new domains or task setups that you have not seen before *but* which you can solve using the skills and techniques you learned in 630. You can think of these as open-ended interview questions where someone wants to hear some kind of general answer with enough specifics to feel confident that if they hired you, you’d be able to work out all the details.

Important note: You are *not* expected to go search for and read about solutions to these (or related) problems—nor should you go out looking. All the material you need to solve these questions has been presented in class. Questions 1 and 4 will provide you with a bit of background reading for context, but you are not expected to learn more about the tasks/background to answer the question. You should think of these as interview-like questions where you are expected to think on the spot.

Please be as specific as you can when describing the inputs and outputs of your system and the features it uses, as it shows how you think the system should work.

As a general hint, we strongly encourage you to think about *all* the different techniques you've learned in the class and ask yourself if there's some way you could apply them to the current problem; pick the best few and you should have a good answer.

Long Question 1 (15 points): Bringing history to life

Historical diaries and letters provide a rich perspective into the events of the day as experienced by people in the moment. Hearing about your success in **Homework 3**, you get hired by the US National Archives to develop a new NLP system that will **extract these personal anecdotes and reflections for specific events in their trove of data, starting with reflections on the US Civil War**. Specifically, they are about to create a set of digitized and transcribed letters and diaries from around the time period of the war and want to create some **big exhibition** with text from these diaries/letters. However, they have *no idea* which content in those letters and diaries relates to the war itself. Further, they've noticed that documents aren't always **entirely focused on the civil war**, e.g., a letter from a soldier to back home may talk about **family matters** in the first half and then go into some reflection on the war, only to talk about home again.

Your boss has high hopes but isn't sure if this project is even possible, so they ask you to take a look at a few letters² and want you to draft a set of annotation guidelines for how you would try to distinguish Civil War-related content within documents. Your guidelines should follow a span-based annotation setup where you **describe when to mark something as related** or not (e.g., how will you deal with unrelated content in the middle of a longer Civil War-related passage) and **which kind of content is included or excluded**. Of course, you cannot read the whole collection so you're not expected to cover all the cases of what is potentially related—later on, archivists will do the annotation and help refine these guidelines—so this initial draft guidelines is just a best-effort. Your boss wants to get a sense of the **complexity** of the annotation task.

² A good transcribed example collection is at <http://omeka.wellesley.edu/civilwarletters/items/browse/>. Here are a few good letters to give you some sense of their diversity:
<http://omeka.wellesley.edu/civilwarletters/items/show/304>
<http://omeka.wellesley.edu/civilwarletters/items/show/405>
<http://omeka.wellesley.edu/civilwarletters/items/show/457>

You are **not expected** to read more than a few letters (which are short).

Long Question 2 (20 points): Better, more private email auto-complete

Email is a lot of work and for busy people, sending email just means receiving more email—a never ending task to respond to. Google’s email service, Gmail, introduced Smart Compose³ to help with this problem by have an NLP system **draft potential replies** to an email. In essence, Smart Compose reads an email and generates a response that a user can edit or send directly in reply. This hopefully saves time for users by making easy to edit replies (or just use the ones that work as-is!). However, not all email services have something like Smart Compose, putting them at a competitive disadvantage.

You get hired at ProtonMail, a smaller but still major email firm that **focuses on privacy and security**. Recognizing the utility of Smart Compose, they want you to build a similar system for their users. You can imagine that Google likely trained its Smart Compose on lots of data from its users. However, ProtonMail specifically asks you to take a different approach. **Only a small percentage** of their users have opted-in to let you train on their data (~1%). The remaining ones will only let you use their own data **to improve their experience specifically** (i.e., you can’t use those users’ data to train a general model but you can use User A’s data to improve the experience of User A). Your boss wants you to get as high a coverage as you can (many emails get a suggested reply) but **doesn’t want them to be bad suggestions**, so you favor precision but are still hoping to get recall as high possible.

Your task is to sketch out what a system would look like for generating potential replies like Smart Compose does in this environment. Describe **which kinds of emails you would respond** to and what strategies/techniques you would use to try generating replies. Be clear about the inputs, outputs, and training data (if any) for your system. Discuss how you would **evaluate your system** and how you would know it’s good before you actually deploy it (e.g., what are you testing, and what are your metrics).

³ <https://www.blog.google/products/gmail/subject-write-emails-faster-smart-compose-gmail/>

Long Question 3 (20 points): Comment your code

Code comments are critical in industry. Highly commented code ensures the code itself is understandable and that new developers who come in can quickly understand what is happening and why. Badly commented code is a liability—what exactly does this line do and what happens if I delete it?

You get hired by IBM who is known for building large software systems that live for a very long time. They are starting to put out more Python code but are still stuck wondering **how good their comments are**. The VP of Engineering hears that you know NLP and has asked you to come up with a way to evaluate the quality of comments in a python file. Evaluating quality can be tricky since a **single comment may describe multiple lines** of code or even a function. Further, comments are directly related to the code—the same text for a “good comment” may be a bad/irrelevant comment if it was put in a different part of the code.

Your task is to sketch out a system for evaluating the quality of text documentation in a python-language code file (e.g., code comments). Since IBM works on huge software systems, they want to know which **files need more documentation**. Your system should provide a **single score** for an input file on its quality of code documentation—but they want some hints on **where in the code the documentation is good/bad**. IBM has lots of money but wants to see that the idea is feasible first so the VP has given you a small seed fund of \$10K USD to do annotation if you need it to prove you can get a prototype system up and running. Describe your system and approach. Describe how you will evaluate your system and convince the VP that it is working.

Long Question 4 (15 + 10 points): Tinker Tailor Soldier Spy

State-level spycraft is still alive and well. Spies can provide valuable information to their country of origin and it is the duty of the originating country to keep them safe. One potential risk to spies is their writing and how they write their secret communications. The particular style in which a spy writes can potentially be linked to other non-anonymous writings—e.g., a spy could send some secret missive that matches the style of an email they sent from their company email, thereby revealing their identity. Stylometric analysis aims to characterize styles and, ideally, help match authors of different texts.⁴

You are an independent consultant who specializes in NLP. A security contracting firm reaches out to you to see if you're interested in a job at a security contractor to help come up with a way to **protect them by preventing others from discovering a spy's identity through stylometric analysis**. Specifically, the job would entail developing a system that takes in some kind of text and then **modifies** its style so that it has a different "style profile" from the author's other writings while still preserving the meaning of the original message. You haven't done much stylometry before, but you figure you can maybe manage after reading a blog post or two. Since you're not directly working for the government, they can't give you *any* **data from actual spies**, so they want you to figure out a way to build the system with other kinds of public data. They do tell you that the average **length of a message is about a paragraph**. Also, since you're an independent contractor, your particular work contract gives you the right to release and code or a description of what you build as well (if you want).

Your task:

1. (15 points for implementation) Describe how you **would build a system to modify the style of a message to avoid having the original messages match the modified one**. To accomplish this task, you will need to specify how you are going to characterize style (e.g., what does it mean for two messages to have the same style quantitatively?) There can be many different stylistic aspects. Then you will need to state how you transform your text to modify each of these, while still preserving the meaning.
2. (10 points for ethics) Describe in detail the **potential risks, harms, and benefits of** different decisions you could take such as taking the job, not taking the job, or whether you release the code (these are just some possibilities). Discuss risks, harms, and benefits in specifics, including how these change if something is *not* done. Finally, discuss how you personally would weigh the risks, harms, and benefits in making your decision on what to do (e.g., would you take the job?) and describe why.

⁴ For a brief introduction, see this blog post <https://programminghistorian.org/en/lessons/introduction-to-stylometry-with-python> You are *not* expected to do more reading than this