CS 4740 (+crosslists): Natural Language Processing **(Due: January 26, 11.59 p.m.)**

Homework #0

Instructors: Tanya Goyal

Course Policy: Read all the instructions carefully before you start working on the assignment, and before you make a submission.

- You should do this assignment **individually**. Do not consult your peers or generative AI engines for any part of this assignment.
- This assignment is designed to test whether you have the necessary background for this course. It should take you around 2 hours to complete this assignment. If you find yourself struggling with a large portion of this assignment, please talk to the course staff about whether this course is appropriate for you.
- This assignment is **ungraded** and there is no submission. However, make sure you compare against the answer key when it is released to self-evaluate your readiness for the course.

Section 1: Math Recap

(10 points)

(1) (1 point) Given two matrices $\mathbf{A} \in \mathbb{R}^{X \times Y}$ and $\mathbf{B} \in \mathbb{R}^{Z \times Y}$ where $X \neq Y \neq Z$. Which of the following operations are invalid? If the operations are valid, what is the dimensions of the final matrix?

(a) $A^T B$
(b) $A^T A B^T B$
(c) $A A^T B B^T$
(d) $A^T A B^T$

(2) (1.5 + 1.5 = 3 points) The joint probability distribution of two random variables X and Y is:

P(X,Y)	X=1	X=2	X=3
Y=1	0.2	0.1	0.1
Y=2	0.1	0.3	0.2

(a) Compute the value of $E[X]$. Hint: First compute marginal probability $P(X)$.

(b) Compute the value of $E[X + Y]$.

(3) (1.5 + 1.5 = 3 points) You have two bags. Bag A has 2 red balls and 3 blue balls, bag B has 3 red balls and 7 blue balls. You randomly draw one ball from a random bag.

(a) What is the probability that the drawn ball is red?

(b) Suppose the drawn ball is red. What is the probability that you drew the ball from Bag A?

(4) (1 + 2 = 3 points) We define the following function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

(a) Prove $f(x) + f(-x) = 1$

(b) Prove $\frac{\partial f}{\partial x} = f(x)(1 - f(x))$

Section 2: Coding (20 points)

In this section, you will implement two kinds of tokenizers: (1) a **sentence tokenizer** that splits a paragraph or document of text into individual sentences, and (2) a **word tokenizer** that splits text into individual words.
With this assignment, you are given a text file `rawtext.txt`. **You can download this file at https://www.cs.cornell.edu/courses/cs4740/2025sp/hw0/rawtext.txt.** Each line in this file contains one paragraph from Wikipedia about a famous person. The document is lowercased. We will use this file for our investigation of different tokenization schemes.

You will not submit code. Just run the code you write and answer the following questions. Include any graphs and analysis you conduct using your code.

(1) First, we will implement a simple sentence tokenizer. For English, we can split a paragraph on certain punctuations like `.?!` etc. (think whether there are other punctuations that need to be included in this set) that occur at the end of a sentence. Write a python function that splits a paragraph into sentences.

(a) (1+ 2 = 3 points) Use your sentence tokenizer to tokenize each paragraph of the `rawtext.txt` file into sentences. Report the average number of sentences per paragraph? Plot a histogram of the number of sentences in a paragraph for the corpus.

(b) (2 points) Can you think of scenarios where naively splitting a sentence on punctuations is not an ideal strategy? What other constraints will help? Discuss.

(c) (2 points) Re-do part (a) using `nltk`'s sentence tokenizer instead of your own implementation. Do you observe any differences? Give 2 examples where sentence tokenization for a given paragraph is different and discuss why.
Instructions to install `nltk` are at: https://www.nltk.org/install.html.

```
import nltk
nltk.download('punkt_tab') # you only need to do this once
paragraph = "x and y went to the market . they bought eggs and bread ."
sentences = nltk.sent_tokenize(paragraph)
```

(d) (1 point) Did you notice any cases where nltk's tokenizer was incorrect?

(2) Next, we will use `nltk`'s word tokenizer to compute some statistics for our text corpus.

(a) (1 point) Iterate through the sentences you tokenized in the previous step (using `nltk`'s sentence tokenizer) and report the mean number of words/sentence.

(b) (2 point) In a number of NLP tasks, one usually removes `stopwords` from a text corpus before using it for downstream tasks. You can view `nltk`'s English language stopwords by:

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

Can you think of reasons why stopwords are removed before performing certain kinds of text analysis? Think of the sentiment classification problem from lecture 1. Are there tasks where stopwords should not be removed?

(c) (2 points) List the 20 most frequent words in the whole corpus (not considering stopwords or punctuations) and their counts. Are there any surprise entries in this list?

(3) We know that each line in the file `rawtext.txt` contains part of a Wikipedia biography.

(a) (1 point) Looking at the top 200 most frequent words in the dataset, can you guess which professions are most frequent in the dataset? Give examples of 3 professions.

(b) (3 points) For each profession you identify, also list 3 other words that co-occur most frequently with the profession word. Two words co-occurring means that they are both mentioned in the same paragraph at least once. E.g. if “actor” is a profession you identify, maybe the most frequently co-occurring words would be “films” or “television”.

(b) (3 points) Let $P(x)$ be the probability that word x occurs at least once in a paragraph. For the profession and related word pairs you identified in part (b), compute $P(<\text{ profession }> \mid <\text{ related word }>)$.