USC Marshall
School of Business

DSO 560 – Text Analytics & Natural Language Processing

Instructor: Yu Chen
TA: Jayden Cho
Final Exam

Due Thursday, December 9th, 8:05pm PST, 90 minutes

No exams will be accepted past 8:10pm PST

Instructions:

- **WRITE ALL ANSWERS ON SEPARATE SHEETS OF PAPER**
- **SCAN EACH PAGE (AS A PDF OR IMAGE) AND SEND TO ME AND THE TA VIA SLACK.**

- **DO ALL SECTIONS.**

ONCE YOU SUBMIT YOUR EXAM, YOU CAN LEAVE CLASS

SHOW ALL WORK TO RECEIVE PARTIAL CREDIT

Short Answer (5 pts, recommended 30 minutes)

Pick 5 of the short answer questions below to answer. Write no more than 2 sentences in your explanation. Each question is 1pt: 0.5pts for the correct answer and 0.5pts for a correct explanation.

1. Using word2vec embeddings, which of the following pairs of words would have the highest similarity score with each other? Why?
    a. happy
    b. matrix
    c. sad

2. Using which encoding scheme would you encounter an **InvalidContinuationByte**? Provide an example of a situation when an **InvalidContinuationByte** error might occur, and why.

3. In a language model's transition matrix T for bigrams, what does the value in T[i][j] represent?

4. Given the following text:

DSO 401: Business Information Systems -- Spreadsheet Applications **(2.0 units)**
DSO 424: Business Forecasting **(4.0 units)**
DSO 428: Essentials and Digital Frontiers of Big Data **(4.0 units)**
DSO 458: Essentials of Business Data Analysis Using R **(4.0 units)**

Write a single regex pattern using capture groups to extract the
    a. Name of the course
    b. Course number (you can assume there are only DSO courses here)
    c. Number of units

5. Provide an example of an input document that an RNN would struggle to predict the next word correctly, and explain why it would have difficulty.

6. For a specific input document as part of a sequence to sequence problem where you are translating from English to Spanish (Neural Machine Translation), you inspect your Transformer model and see that it has generated the following matrix for attention scores:

0.5	0.2	0.3	0.0
0.1	0.7	0.1	0.1
0.5	0.0	0.4	0.1
0.0	0.1	0.0	0.9

a. What is the sequence length of the input? How do you know?
b. Provide any hypothetical scenario that might explain the values in the third row of the matrix (0.5, 0.0, 0.4, 0.1).

7. Which of the following lines of code should you select to read into memory a 500GB text file? Explain why.
    a. df = pd.read_csv("filename.txt")
    b. file = open("filename.txt")

Naïve Bayes (3 pts, recommended 15 minutes)

You are a data scientist at Zendesk implementing a **Naïve Bayes NLP model to classify urgent customer support ticket summaries from your customer support agents.** This model will be used to quickly alert customer service to follow up with customers.

You've collected a small dataset below:

Urgent Ticket:

1. Credit card stolen
2. User passwords leaked
3. Refund stolen from customer

Not Urgent Ticket:

1. User forgot password
2. Customer needs refund
3. Stolen merchandise
4. Customer doesn't speak English

Note – perform any text preprocessing you deem necessary. Please state your assumptions.

Perform any preprocessing and grouping you deem appropriate, and calculate the following probabilities, assuming a Naïve Bayes classifier (2 pts):

i. The **prior probability** for an urgent ticket **(0.5pts)**
ii. Calculate the **posterior probability** for an Urgent ticket that says "User password stolen" (**2pts**)

Determine if the P("refund") and P("customer") are independent from each other (**0.5pts**)

Vectorization and Similarity (3 pts, recommended 15 minutes)

You work as a data analyst for Pinterest. Your team is tasked with identifying customers' clothing preferences. This is what 3 customers wrote in a recent survey for their anticipated next clothing purchase:

User A: work dress **User B:** artsy fur dress **User C.** cute coat

These are your definitions for **Term Frequency** and **Inverse Document Frequency**:

TF = n(t,d) **IDF =** 1 / df(t) + 1

n(t,d) is the number of times term *t* appears in document *d*
df(t) is the document frequency of term *t*

Please group any tokens you feel should be collocated together.

a. Generate **TF-IDF document vectors** (you may write them as a matrix or table) for each of the 3 users (**2pt**).
b. A new clothing item has been tagged as **cute dress**. Assuming **TF-IDF vectorized** documents and **Cosine Similarity**, should we recommend this for **User A** or **User B**? **(1pt).**

N-Gram Language Models (3 pts, recommended 15 minutes)

Given the following documents:

1. I went home late.
2. They eat dinner late.
3. I eat dinner at home.
4. They went late.

A. Perform lemmatization and stopword removal. For this exercise, consider the words **"the", "to", and "at"** as stopwords. Then construct the transition matrix for this corpus. (**2 pts**)

B. Using a **bigram language model** what is the likelihood and perplexity for the sentence ***I went to dinner late?*** (**1pt**)

True/False (5 pts, recommended 20 minutes)

Pick 5 of the statements below, indicate if it is true or false. **In both cases (true or false), explain your reasoning in a brief sentence. Each question is worth 1pt: 0.5pts for the correct answer, 0.5pts for explanation.**

A. Assume punctuation and stopwords like "the" are removed during text preprocessing. The cosine similarity of the following documents would be 1:
   a. "The cats love dogs."
   b. "The dogs love cats, the cats love dogs."

B. If you wanted to visualize your documents on a scatterplot, after vectorization, you could set the number of components in your dimensionality algorithm to 2 or 3.

C. Using a BERT model, the embedding vector for the token "brief" would be identical for an input document "the lawyer prepared a detailed legal brief" and for another input document "the speech was brief and only consisted of a few sentences".

D. Each row of the Hidden Markov Model's emission matrix should sum to 1 and represent the likelihood of transitioning from one word at sequence step i to another word at sequence step i + 1.

E. If you do not use word boundaries in regex for short, common phrases are "bro", "thor", "the", etc. you will increase the number of false negative results.

F. If you are implementing a text preprocessing pipeline for a search query engine where you want to maximize recall, you would use stemming.

G. Your RNN is more likely to suffer from the vanishing gradient problem if the input sequence length is very short and most of the weight values are over 1.