USC Marshall School of Business

DSO 560: Text Analytics + NLP

Winter 2023

FINAL EXAM

Date Given: Feb 28, 2023 Due Date: March 10, 2023

==========

There are 8 problems in this assignment. Total Points = 25

==========

Problem#1 (4 points)

Positional Encoding: The '*Attention is all you need*' paper describes position encoding as follows.

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

Suppose the following data is given:

- Positional embedding size = 8 (d_{model} as described in the above formula))
- Max position size = 4 (i as described in the about formula)

Write Python code that computes 'Positional Encoding' for all positions. You should expect to get the following answer. The table below shows the 4 positional vectors (P(0), P(1), P(2) and P(3) with size 0f 8.

```
[[0.    0.    0.    0.    0.    0.    0.    0.   ]
 [0.841 0.995 0.01  1.    0.    1.    0.    1.   ]
 [0.909 0.98  0.02  1.    0.    1.    0.    1.   ]
 [0.141 0.955 0.03  1.    0.    1.    0.    1.   ]]
```

Compute the cosine distance between 1st and 3rd positional vectors. You should expect to get the following answer.

```
Cosine distance with positional embeddings
p1->p3
0.053330553292434746
```

Problem#2 (3 points)

Given the embeddings of the following words, compute the 'self-attention' matrix

Suppose the embeddings of the following 3 words are as follows. The embedding size is 4.

```
word_1 = np.array([1, 0, 1, 0])
word_2 = np.array([0, 2, 2, 2])
word_3 = np.array([1, 1, 1, 1])
```

Attention matrix can be computed using the Q,K,V matrices. The values of Q,K,V are computed using the training period. Let us assume that the values of Q,K,V matrices computed after the training period are as follows.

```
W_Q = np.array([[0, 0, 1],
                [1, 1, 0],
                [0, 1, 0],
                [1, 1, 0]])

W_K = np.array([[1, 0, 1],
                [1, 0, 0],
                [0, 1, 0],
                [1, 0, 1]])

W_V = np.array([[1, 0, 1],
                [1, 1, 0],
                [0, 1, 1],
                [0, 0, 1]])
```

Compute the attention vector of all the 3 words. Your answer should be as follows.

```
[[1.83205655 2.89785022 3.36495339]
 [1.99996943 3.99371005 3.99683974]
 [1.99706398 3.88587728 3.94147063]]
```

Problem#3 (3 points)

The Huggingface Transformer library's API will allow you to access many NLP models.

- GPT (Generative Pre-Training) Models
- BERT (Bidirectional Encoder Representations Transformer) Models

The following URL will take you to the Huggingface's Transformer website.
https://huggingface.co/transformers/

Problem#3.1: Using the Huggingface's Transformer library and the 'sentiment-analysis' pipeline, analyze the sentiments of the following sentences.

I like NLP course.
I hate when my computer crashes.

Problem#3.2: Using the Huggingface's Transformer library and 'zero-shot-classification' pipeline, classify the following sentence in one of the three given classification categories.

Text: Los Angeles Clippers is a good basketball team
Given Classification categories = sports, politics, education

---

Problem#3.3: Using the Huggingface's Transformer library and 'text-generation' pipeline, complete the following sentence.

In this month, the stock market will

Problem#3.4: Using the Huggingface's Transformer library and 'fill-mask' pipeline, fill in the blanks.

Math course will teach you about <mask> topics

Problem#3.5: Using the Huggingface's Transformer library and 'ner' (Name Entity Recognition) pipeline, identify name, organization, and place.

Tim Cook is the CEO of Apple located in San Jose.

Problem#3.6: Using the Huggingface's Transformer library and 'question-answering' pipeline, let the system find the answer to the following question in the given context.

Question: In which state Los Angeles located
Context: Los Angeles is in California

Problem#3.7: Using the Huggingface's Transformer library and 'summarize' pipeline, summarize the following text.

Text:

```
Australia was celebrated for its initial response to the Covid-19 pandemic,
and for getting its economy more or less back on track long ago.  But with
that security has come complacency, particularly in the federal government,
which failed to secure enough vaccine doses to prevent the regular "circuit
breaker" lockdowns that come every time a handful of cases emerge, or even
the longer restrictions that Sydney is experiencing now. Australia's
borders, controlled by strict quarantine measures, have been all but shut
for more than a year.  Now Australians, who basked in their early successes,
are wondering how much longer this can go on.
```

Problem#4 (3 points)

Analyze the embeddings of the word ‘bank’ in the following 2 sentences.

Sentence 1: I went to a *bank* to deposit money.
Sentence 2: I sat near a *bank* of a river.

Download the embeddings of all the words in the above 2 sentences for the following 2 models.

- Glove embeddings
- BERT embeddings from Hugging face Transformer web portal

Compute the Euclidian distance between the Glove and BERT embeddings for the word ‘bank.

Expected answer:

- Glove embedding: Euclidian distance between the embeddings for the word ‘bank’ used in 2 sentences $= 0$

- BERT embeddings: Euclidian distance between the embeddings for the word ‘bank’ used in 2 sentences $\neq 0$

Problem#5 (3 points)

Deep Learning has revolutionized Natural Language Processing (NLP). Personal assistant software like Siri (Apple's iPhone), Google Assistant (search engines), Amazon's Alexa, and robots use text and audio NLP to understand the user's intent and provide an accurate response. Language Models have also been developed for dialog applications which allow a form of intelligent conversation between a human and a computer.

Deep Learning technology has facilitated development of many Natural Language (NL) models like BERT (Google/2018), GPT-1/2/3 (Open AI/2018 - 2019). In May 2021, Google introduced a new model called LaMDA for dialog applications. This new Deep learning Language Model is trained on enormous amount of data with trillions of parameters. This raises the following questions:

- Are the new models providing more accuracy in its answers than previous models?
- Is there a limit to the size of data and parameters used to build such models?
- What does future hold for Language Models?

Language Models represent the knowledge embedded in the data on which they are trained. If the corpus on which the model is trained is biased, answers given by the model would also be biased.

A recently published research paper, "On the Dangers of Stochastic Parrots: Can Language Models be Too Big?" was written by faculty members from the University of Washington and by Google researchers. Since this paper addresses the risk of building bigger Language Models, Google management was hostile and the Google researchers who participated in writing it were fired after publication.

This paper analyzes the risks of building bigger models. There are many risks, but main one is getting incorrect answers from models trained on biased data. Google's status in society projects an authority onto the answers given by the Google search engines. Therefore, incorrect information provided by the models can be harmful to society. This leads to asking the obvious question: What is the future of Language Models?

==========

I have provided the paper "On the Dangers of Stochastic Parrots: Can Language Models be Too Big?" in PDF format. Professor Emily M. Bender of University of Washington gave a talk on this subject at "The Alan Turing Institute" on July 8, 2021. The summary of this talk is also provided in PDF format. The YouTube's video on the Professor Bender's talk is available at the following link.
https://www.youtube.com/watch?v=N5c2X8vhfBE&t=75s

Your job is to read the paper written by Professor Bender and watch the YouTube video. Summarize the content of paper in one or two pages (just a few paragraphs).

Problem#6 (3 points)

The “worldfloras.csv” file contains information about all the countries in the world. This file is in ‘comma separated’ format.

	A	B	C	D	E	F	G
1	Country	Latitude	Area	Populatior	Flora	Endemism	Continent
2							
3	Afghanistan	30	636	14.3	3000	0.27	Asia
4							
5	Albania	42	29	3	3200	0.008	Europe
6	Algeria	35	2382	21.3	3139	0.08	N.Africa
7	Andorra	42	0.5	0.034	1000	0	Europe
8	Angola	25	1247	8.5	5000	0.25	Africa
9	Antarctica	85	14000	0	2	0	Antarctica
10	Argentina	45	2777	30.1	9000	0.27	S.America
11	Australia	25	7682	15.5	23000	0.8	Australia
12	Austria	48	84	7.5	3000	0.012	Europe
13	Bahrain	26	0.66	0.4	175	0	Asia
14	Balearic Islands	40	5	0.62	1400	0.067	Europe
15	Bangladesh	23	144	98.5	5000	-1	Asia
16	Belgium	52	31	9.9	1700	0	Europe
17	Belize	14	23	0.16	3240	0.046	C.America
18	Benin	5	113	3.9	2000	0.006	Africa
19	Bhutan	30	47	1.4	5000	0.12	Asia

Read the ‘worldfloras.csv’ file in Python. The data in this file is used for problem #1 - #5. Use Python ‘re’ library for regular expressions.

Display the country names that have the following characteristics in their name.

- The letter 'z' as the 2nd character
- The letter 'h' as the 5th character
- The letter 'l' as the 8th character
- The letter 'k' as the 12th character

Use Python ‘re’ library for regular expressions.

Problem#7 (3 points)

Compute the TF-IDF vectors for the following corpus.

```
corpus = [ 'This is the first document.',
     'This document is the second document.',
     'And this is the third one.',
     'Is this the first document?']
```

There are 2 definitions of TF-IDF vector – (1) Standard definition (2) Alternatedefinition.

Use the 'CountVectorizer' and 'TfidfTransformer' packages from 'Scikit-Learn' library and compute the values of TF-IDF using the 'Alternative Definition' of TF-IDF.

- Standard Definition of TF-IDF
  - $tf(t,d) = f_{t|d}$
  - $idf(t,D) = log_e\left(\frac{N}{|\{d \in D: t \in d|}\right)$
  - $tf.idf(t,D) = tf(t,d).idf(t,D)$
- Alternative Definition of TF-IDF used in SKLearn TD-IDF Vectorizer
  - $tf(t,d) = f_{t|d}$
  - $idf(t,D) = log_e\left(\frac{N+1}{|\{d \in D: t \in d|+1}\right) + 1$
  - $tf.idf(t,D) = tf(t,d).idf(t,D)$
  - After the values TF-IDF vector are computed using the new definition
    - Next, Compute the Norm of the TF-IDF Vector

Do not remove stop words from the documents. You should expect to get the following values of the TF-IDF vector for all 4 documents.

```
array(['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third',
       'this'], dtype=object)
```

```
[[0.         0.         0.51184851 0.        ]
 [0.46979139 0.6876236  0.         0.46979139]
 [0.58028582 0.         0.         0.58028582]
 [0.38408524 0.28108867 0.26710379 0.38408524]
 [0.         0.         0.51184851 0.        ]
 [0.         0.53864762 0.         0.        ]
 [0.38408524 0.28108867 0.26710379 0.38408524]
 [0.         0.         0.51184851 0.        ]
 [0.38408524 0.28108867 0.26710379 0.38408524]]
```

Problem#8 (3 points)

Using the Singular Value Decomposition (SVD) principals, perform the Latent Semantics Analysis (LSA) of the following 6 documents. Divide these documents into 2 different topics.

```
corpus = [
          'French revolution Napoleon',
          'French revolution Louis',
          'French philosopher Voltaire',
          'Catholic church',
          'Pope Catholic church',
          'Martin Luther Catholic church'
         ]
```

What are the topics on which LSA has divided these set of documents?

Plot all the 6 Truncated vectors with the following specifications.

- X-axis: Truncated DTM (Document Term Matrix)/Topic-1 values
- Y-axis: Truncated DTM/Topic-2 values

Compute the ‘cosine similarity’ and ‘cosine distance’ using the Truncated SVD.