University of Glasgow

Wednesday, 20 May, 09:15 BST
(24 hour open online assessment – Indicative duration 1.5 hours)

DEGREES of MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

TEXT AS DATA (H)
COMPSCI 4074

(Answer 3 out of 4 questions)

This examination paper is worth a total of 60 marks.

1. This question is about tokenization and similarity.

(a) This part concerns processing text. Consider the input string:

[He didn't like the U.S. movie "Snakes on a train, revenge of Viper-man!", now playing in the U.K.]

Provide a tokenised form of the above string. Identify and discuss **two** elements of the above string that present ambiguities. Justify your tokenisation decision for each. [3]

(b) Consider the two tokenized documents:
S1: [a, woman, is, under, a, mayan, curse]
S2: [a, woman, sees, a, mayan, shaman, to, lift, the, curse]

Create a Dictionary from the two documents above (S1 and S2) with appropriate ordering. Give your answer in the form of a table with ID and token. Discuss the following properties of the dictionary and provide reasons for the decision: 1) what is included in the dictionary and 2) the order of the dictionary. [3]

(c) Critically evaluate the Bag-of-Words (BoW) model as a term weighting feature model for documents. Discuss its strengths and give **three** weaknesses of the model and propose a modification that addresses each. You should relate each to Sci-kit Learn vectorizers and their important parameters. [4]

(d) You are measuring the similarity between two molecular compounds for drug discovery research. They have been processed to create a series of unique structural 'fingerprints' and a one-hot encoding of the compounds is created. A compound has tens of thousands of fingerprints on average and all the compounds are approximately the same size. Also, most of the compounds in the dataset share more than 90% of fingerprints in common. A lab partner suggests using Jaccard overlap to measure the similarity between compounds. First, critically discuss why Jaccard is or is not appropriate for this task and the challenges it presents. Second, propose and justify a change to **both** the *representation* and *similarity measure* to address them. [6]

(e) A colleague attempts to call `fit()` on a SKLearn `TFIDFVectorizer` for a collection of data with tens of millions of documents (a large collection of Reddit). The vectorizer takes a long time and then the process crashes with a memory error. Describe the behaviour of `TFIDFVectorizer` and why it is crashing. Propose an alternative vectorization method and compare and contrast it with `TFIDFVectorizer`. [4]

2. This question is about language modeling and classification.

(a) This task involves developing an order error corrector for a popular burger chain, 'out-and-in burger'. Below is a table of five separate order interactions transcribed from a mobile app.

forget it i wanna eat a hamburger
no i wanna eat a hamburger
i would like to eat breakfast
i would like to eat a cheeseburger and a beer
would you like fries with that

Table 1: Five interactions for a burger restaurant ordering system.

Sample text collections statistics for a bigram model are below:

- $V = 22$ unique words (including reserved tokens)
- $N = 45$ tokens, including padding

(i) Use the text provided in Table 1 above to compute word unigram probabilities. In a list or table format complete the probability table with Laplace smoothing that has $K = 0.5$. Show your workings. Discuss the impact on the probability values of increasing or decreasing the value of K. Describe the effect of K when these probabilities are used in a spelling (error) correction task.

Word	Unigram Probability
breakfast	
beer	
hamburger	

[5]

(ii) A larger collection of restaurant ordering data is collected. It has the following statistics: $N = 73194$, $V = 1996$ from a total of 8565 documents (utterances).
Compute the bigram probability of the following sequence:

```
[i might like a cheeseburger]
```

with Stupid Backoff smoothing with default values. Collection statistics for the required terms are provided below. Show your workings, including each bigram's probability. Describe how and why a smoothing method is used here.

[6]

Term	Count
i	2815
might	4
like	1522
a	1051
cheeseburger	3
$\langle s \rangle$ i	1926
i might	0
might like	1
like a	49
a cheeseburger	3
cheeseburger $\langle \backslash s \rangle$	2

(b) Compare and contrast the APIs for SKLearn Transformers (e.g. Count or TF-IDF) and Classifiers/Predictors (e.g. NaiveBayes, LogisticRegression). Include descriptions of their key interface functions with descriptions of their behaviour. Discuss how they are used together to solve machine learning tasks on text. [3]

(c) Below is a snippet of code to vectorize and classify text with Scikit-learn. Assume that `tokenize_normalize` and `evaluation_summary` have been defined, as we did in the labs. The input data has been pre-processed into a vector of unnormalized text documents (each a single string).

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
# Data processing
data = ... # Loads a vector of raw text documents
train_index = int(len(data) * 0.1)
train_data = data[:train_index,:]
validation_data = data[int(train_index*0.2):,:]
test_data = data[train_index:,:]
# Assume corresponding labels for each data subset
train_labels, test_labels, validation_labels = ...

# Vectorization
one_hot_vectorizer = CountVectorizer(tokenizer=tokenize_normalize,
    binary=True, max_features=20)
one_hot_vectorizer.fit(train_data)
train_features = one_hot_vectorizer.transform(train_features)
validation_features = one_hot_vectorizer.fit_transform(validation_data)
test_features = one_hot_vectorizer.transform(test_data)

# Classification
lr = LogisticRegression(solver='saga', max_iter=500)
lr_model = lr.fit(train_features, train_labels)
evaluation_summary("LR Train summary",
    lr_model.predict(train_features), validation_labels)
lr_model = lr.fit(validation_features, validation_features)
```

```
evaluation_summary("LR Validation summary",
    lr_model.predict(validation_features), validation_labels)
lr_model = lr.fit(test_features, test_labels)
evaluation_summary("LR Test summary",
    lr_model.predict(validation_features), test_labels)
```

Copy and paste the code above and fix its mistakes. Although there may be more, discuss **three** important mistakes with their consequence, one from each section (data processing, vectorization, classification). [6]

3. This question is about word embedding models and Natural Language Processing.

(a) Compare and contrast static word embeddings with contextual embedding models. Discuss the trade-offs between them for downstream tasks. [4]

(b) Using your knowledge of the self-attention mechanism used in Transformers, answer the following questions considering the following sentence:

S1: [The president of the European Union spoke]

(i) For each of the following 'query' words: [president, Union, spoke]; Produce indicative (made up) attention weights (rounded to two decimal places) for the other words in the sentence. Provide a justification for your weights for each query word.

Word	Att: "president"	Att:"Union"	Att:"spoke"
The			
president	—		
of			
the			
European			
Union		—	
spoke			—

[3]

(ii) Discuss the role of different layers in Transformer models (like BERT) and how they relate to steps in a traditional NLP pipeline. [2]

(c) In this question we explore what can be done when faced with a completely "alien" scenario. Klingon is a language originating from TV series Star Trek. Many classic works such as Hamlet, Much Ado About Nothing, Tao Te Ching, and Gilgamesh have been translated by hand to Klingon. It is studied and formalized by the Klingon Language Institute (KLI) and was designed to be dissimilar from English. Below are some sample Klingon-English translations.

Klingon	Approximate English Translation
taH pagh, taHbe'	Whether to continue, or not to continue [existence]
bIpIv'a'	How are you?
munglIj nuq	Where are you from?
Huch 'ar DaneH?	How much is this?

Figure 1: Sample sentences in Klingon

You will use your knowledge of text processing and NLP to understand what is being said by Klingons and the actors portraying them on TV.

(i) Describe the process for pre-training BERT on Klingon. Briefly describe what is required and any changes needed for the model. Be sure to include discussion of tokenization considerations and training objectives. [3]

(ii) We want to identify when an actor makes a mistake (uses an incorrect word) when reciting a Klingon sentence from the script. Describe how to apply BERT to *identify* and to *suggest fixes* for likely mistakes. [4]

(iii) We want to incorporate embeddings into an existing Klingon intent classifier in order to distinguish between 'romance' and 'anger' utterances. Early experiments tried naïve averaging and maxing, but these were not effective. Propose an alternative method to aggregate token-level embeddings into a single representation. Justify why your proposed approach may be effective. [4]

4. This question deals with Information Extraction and its applications.

You work as a data scientist for Pear, a (fictional) large consumer smart device company. You are given a spreadsheet of product data (including product names, model numbers, descriptions, and technical specifications) with several hundred new product releases. You are also given a collection of several million discussions collected from social media websites. Your task is to use Information Extraction (IE) and sentiment analysis to analyse consumer reactions to the new products. The output should be an overall sentiment analysis summary as well as a detailed breakdown of important issues discussed for each product. You will use your knowledge of supervised and unsupervised text processing and machine learning to design an appropriate solution to this task.

An incomplete partial product row and a corresponding sample of social media data are provided below:

Product ID	Part number	Product name	Product description	Price	Dimensions	Weight	...
28151234	3201	Pearl Smart Grilling Hub	Your secret ingredient to perfect BBQ...	99 GBP	14CM H X 14CM W X 7CM D	500 grams	...

Table 2: Sample product spreadsheet row

Post	Source	Text	Author	...
1	http://url1	I just received my Pearl Smart Grilling Hub and wanted to share my initial thoughts. I had an earlier Pear Semi-Smart Grill model that this replaced. Here goes... The Hub is small, magnetic, and has giant numerals to display the numbers. All good! It required no less than 5 firmware updates, which took about 40 minutes. It connected to my phone, but then it got finicky. Temps didn't appear, the app crashed. It's incredibly slow to see your history. The Hub is small, magnetic, and has giant numerals to display the numbers. All good! It required no less than 5 firmware updates, which took about 40 minutes. It connected to my phone, but then it got finicky. Temps didn't appear, the app crashed. It's incredibly slow to see your history.	author1	...
2	http://url1	Update: they pushed an update in early March that seems to have fixed all the issues. Currently working great!	author1	...
3	http://url1	Just got mine last Friday, it did about 30 mins of updates and still rubbish.	author2	...

Table 3: Sample social media data for product

(a) Task definition - Define the output for the extraction task in detail. Provide a sample output schema with fields for at least two tables: **Summary** and **Details**. For each field in the schema give its definition and why it is important for the task. Give an illustrative instance that represents the output of information extraction on the sample product discussion data provided in Table 2. Briefly discuss your design and justify how it meets the target task requirements. [5]

(b) System design - Provide a detailed design of the NLP extraction pipeline with the required key sub-components and sub-tasks. For each main component describe its input, its core function, and output. Characterise the task type and select an appropriate model, providing a rationale for each. Discuss modifications needed to adapt 'off-the-shelf' models or train new models and the data required. Critically analyse your design and discuss alternatives and their trade-offs. [5]

(c) Entity handling - One key challenge is understanding product entity mentions. Compare and contrast the tasks of coreference resolution with entity linking. Discuss how each is used in the product extraction task using illustrative examples from the example text provided in Table 3 above. [4]

.

(d) Evaluation - Develop an evaluation plan to measure and improve the effectiveness of the extraction system. Describe an appropriate experimental setup for the system evaluation. Define and describe a "top-level" main evaluation measure for the system overall. Provide appropriate evaluation measures and methods for **three** of the system's sub-modules. [6]

END OF QUESTION PAPER