Mandatory Assignment #3

Text Analytics

Weifang Wu

ww.digi@cbs.dk

Copenhagen Business School, Denmark

Deadline: {check your digital exam for exact date and time}

Instructions

Please note that you have to upload the solutions before the deadline to the digital exam http://exam.cbs.dk/.

Please use Python 3 for answering questions in Part 2. It is also a good practice to use comments extensively in your code, so that it will be easy for other people to understand it.

Finally, in case Python code is involved in answering a question in the assignment, you can submit a jupyter notebook containing all the code and report as markup in one file. Alternatively you can also submit python code (with .py extension) and the report as two separate individual files.

Part 1: Simple Problem

1. An IR system returns 10 relevant documents and 8 non-relevant documents. There are a total of 25 relevant documents in the collection. What is the precision of the system on this search, and what is its recall?

2. Draw the inverted index that would be built for the following document collection.

```
Doc 1: new home sales top forecasts
Doc 2: home sales rise in july
Doc 3: increase in home sales in july
Doc 4: july new home sales rise
```

3. Consider two documents A and B whose Euclidean distance is d and cosine similarity is c (using no normalization other than raw term frequencies). If we create a new document A' by appending A to itself and another document B' by appending B to itself, then:

a. What is the Euclidean distance between A' and B' (using raw term frequency)?

b. What is the cosine similarity between A' and B' (using raw term frequency)?

c. What does this say about using cosine similarity as opposed to Euclidean distance in information retrieval?

4. Suppose we run the SNOWBALL algorithm on the text below to attempt to extract the FOUNDER-OF relation. Which of the patterns below will extract at least one correct example of that relation without extracting any incorrect ones (select all that apply)?

a) **ORG**, founded by PERSON
b) **ORG**, PERSON
c) founders of **ORG**, PERSON
d) PERSON of **ORG**

ORG entities are **bold**, and PERSON entities are underlined. You can assume that all of the patterns are well formed SNOWBALL patterns.

Source text:
Microsoft, founded by Bill Gates, produces both computer software and personal comput- ers. The founders of **Google**, Larry Page and Sergei Brin, developed an advanced search experience. And Mark Zuckerberg, founder of **Facebook**, crafted a new communication platform. And, usage exists between them: indeed, Bill Gates is a user of **Google** search, and Larry Page of **Microsoft** products such as Word. Bill Gates of **Microsoft**, Larry Page and Sergei Brin of **Google**, and Mark Zuckerberg of **Facebook** were all pioneers of today's technology.

Correct examples:
(**Microsoft**, Bill Gates)
(**Facebook**, Mark Zuckerberg)
(**Google**, Larry Page)
(**Google**, Sergey Brin)

5. Suppose that we run 3 queries in a QA system for its evaluation. For the 1st query, the system returns 10 candidate answers and the 4th is the first correct answer. For the 2nd query, the system returns 5 candidate answers and the 2nd is the first correct answer. For the 3rd query, the system returns 3 candidate answers and none of them is the correct answer. What is the mean reciprocal rank (MRR)?

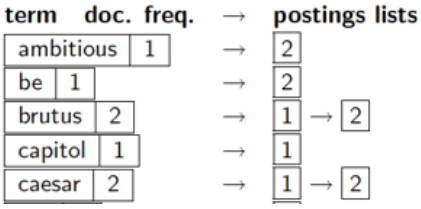
Part 2: Python Programming

In this exercise you are going to build up a simple Boolean retrieval system by applying inverted index. The dataset consists 1,000 wine reviews including two columns doc_id and doc_content. All the texts are in English. The dataset is provided in the form of csv file (*wine-reviews.csv*) along with this assignment. You can use pandas python library to load the given data from the csv file as shown in the following sample coding.

```
# load csv file
import pandas as pd
df = pd. read_csv('wine-reviews.csv')
print(df.head ())
print(df.tail ())
```

Your task is:

First, **to build the inverted index for the corpus**, that is, the 1,000 wine reviews. The inverted index should be like the one we have discussed in class. For example, the one as the following. Before this step, you need to preprocess your text by tokenizing, lowercase, etc. Your terms should be alphabetically ordered. You may consider to use the sorted function in Dictionary.



Second, **to write a query function to handle simple queries of AND**. Please keep in mind to handle the situation that there are no results returned.

Third, **to test your query function** by doing the following queries:

- *rose aroma*
- *aroma acidity flavor*
- *hotel*