FACOM- UFU
Teacher: Wendel Melo

Fourth paper on Information Organization and Retrieval 2023-1

Description

This work consists of calculating and plotting the graph and average precision and revocation for a reference collection.

Only **one** program developed in Python 3 that performs the requested task should be submitted. The program should only use the standard Python 3 libraries, i.e. the libraries that come with the default installation of the language interpreter. Use of the *numpy* and *matplotlib* libraries is also permitted.

The work must be done **individually** and the code generated must be attached to the respective task in the
MS Teams by the deadline indicated.

Important note: if copying or any kind of cheating between assignments is detected, everyone involved will be penalized with a zero grade. So think carefully before asking to copy your classmate's work, as they could be punished too!

Before you start developing, make sure you've understood the slides on evaluating recovery. After studying the slides, come back and read the description again. To do this work, check out the *notebook* and/or the video about the lesson on plotting graphs with *Matplotlib* (the remote task was given in the replacement lesson on 03/11/2023).

Entering the program

Your program should receive an input file (the name of which is passed in from the command line) specifying ideal answers from a fictitious system for reference queries. To understand the input file, let's take the exercise from the recovery assessment slides as an example:

Exercise: Consider a reference collection. Suppose that the sets R1, R2 and R3 of documents relevant to the queries q1, q2 and q3, respectively, have been determined by a group of experts. The sets R1, R2 and R3 are given as follows:

R1 = {d3, d7, d12, d13, d26, d68}
R2 = {d1, d2, d9, d24, d51, d52, d70, d82}
R3 = {d2, d3, d6, d16, d20}

Consider that a new retrieval algorithm called XYZ has just been designed. Suppose that this algorithm returns the following document rankings for queries q1, q2 and q3:

Query q1 = {d1, d9, d26, d15, d2, d10, d74, d68, d32, d3, d53, d39, d56, d11, d4}.

Query q2 = {d3, d7, d8, d9, d19, d16, d37, d24, d20, d80, d67, d50, d46, d51, d29}. Query q3 = {d2, d30, d25, d3, d9, d7d6, d39, d75, d19, d26 d16, d20, d51, d1}.

Construct the precision versus recall graph for each of the queries and the graph with the average precision per recall of the XYZ system.

Thus, the first line of the input file contains the number *n of* reference queries (in the previous example, *n* = 3). The following *n* lines specify the ideal outputs for each of the reference queries, where the *i-th* line specifies the ideal output for query *i*. The ideal answer for each query will be contained entirely on one line, with the documents separated by spaces. Then, the next *n* lines specify the answer obtained by the system for each of the reference queries, where the i-th line specifies the system's output for query *i*. The system's response to each query will be contained entirely on one line, with the documents separated by spaces. For the previous example, we would have the following input file:

```
3
3 7 12 13 26 68
1 2 9 24 51 52 70 82
2 3 6 16 20
1 9 26 15 2 10 74 68 32 3 53 39 56 11 4
3 7 8 9 19 16 37 24 20 80 67 50 46 51 29
2 30 25 3 9 76 39 75 19 26 16 20 51 1
```

sample input file reference.txt

Note that in black, on the first line, we have the number of reference queries (3). In blue, the ideal responses and finally, in red, the system responses. The name of the input file must be received by the command line. So, assuming the input file is called *referencia.txt* and your program is called *avaliacao.py*, we'll call your program by doing:

> python3 evaluation.py reference.txt

Leaving the program

Your program should generate an output file called ***media.txt*** with the system's average precision at each of the 11 standard recall levels (0%, 10%, 20%, ..., 90%, 100%). Simply store the 11 precision values in the output file (11 numbers in the output file and nothing else). For example, for the following precision per revocation table:

Revocation	**Precision**
0%	64%
10%	64%
20%	50%
30%	48%
40%	40%
50%	40%
60%	32%
70%	30%
80%	0%
90%	0%
100%	0%

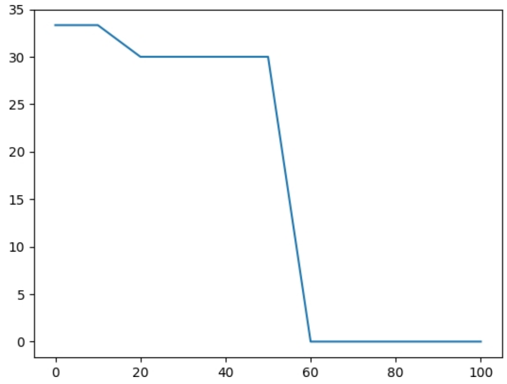
We will have the following output file:

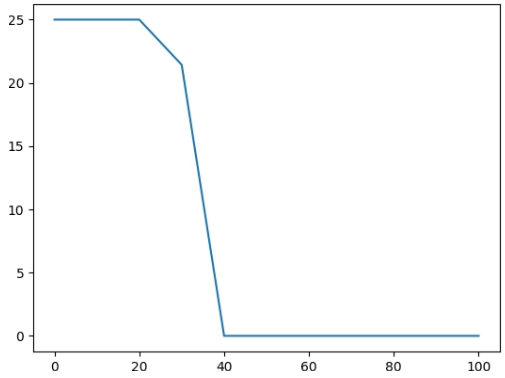
```
0.64 0 . 64 0 . 5 0 . 48 0 . 4 0 . 4 0 . 32 0 . 3 0 0 0
```

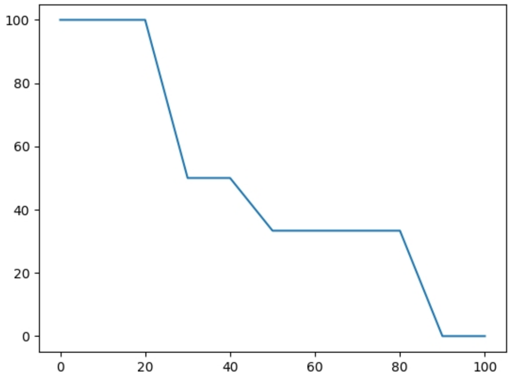
sample output file media.txt

Your program should also generate a precision per revocation graph (at the standard revocation levels) for each of the reference queries and a graph with the system average. **Consult the lesson material to understand how to perform the calculations correctly and to learn how to plot graphs with matplotlib**. The graphs can be plotted on screen (you don't need to save them as a file). Note that the interpolation rule must be used to calculate the standard revocations.

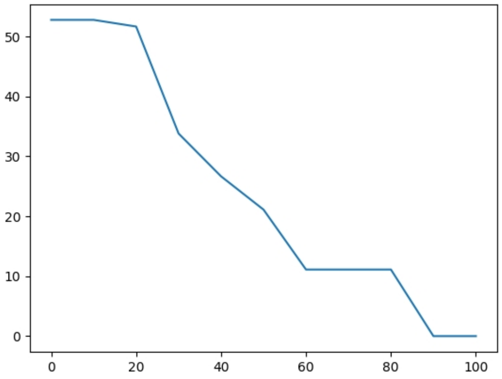
For the input of the example described here, we would have as output:


35
30
25
20
15
10
5
0
0
20
40
60
80
100


25
20
15
10
5
0
0
20
40
60
80
100


100
80
60
40
20
0
0
20
40
60
80
100

Averages:


50
40
30
20
10
0
0
20
40
60
80
100