Information Retrieval Task
ICS2205 Web Intelligence

Joel Azzopardi

October 28, 2023

This document contains the details for the individual Information Retrieval Task for ICS 2205/2230 project. This is which is marked out of 100%, however it is equivalent to 20% of the total mark for this unit (it constitutes half of the 40% individual project component).

The **deadline** for this project is **Monday 8th January 2024 at 1200**. Deliverables and attached plagiarism form must be uploaded on the VLE. Projects submitted late will be penalised or may not be accepted.

1 General Specifications

This project consists of a single task that is divided into a number of subtasks. You will need to create a Jupyter[1] notebook for the task, and the notebooks should also include inline comments and markdown detailing important aspects of the code.

2 Task Specification: Information Retrieval

For this task, you are to use the *WES* dataset available from `https://pikes.fbk.eu/ke4ir` (and also on VLE) to build a simple Information Retrieval engine that uses the Vector Space model to find documents related to a user query.

You need to use the following files from the dataset:

- The document collection – `https://knowledgestore.fbk.eu/files/ke4ir/docs-raw-texts.zip`; and
- The set of queries – `https://knowledgestore.fbk.eu/files/ke4ir/queries-raw-texts.zip`.

Both of these files are being made available on VLE.

[1] `https://jupyter.org/`

An IR engine consists of 2 parts – the document indexing part, and the querying component.

For the document indexing part, you need to implement these process steps:

1. Parse the document to extract the data in the XML's $< raw >$ tag (**10 marks**);
2. Tokenise the documents' content (**5 marks**);
3. Perform case-folding, stop-word removal and stemming (**20 marks**);
4. Build the term by document matrix containing the *TF.IDF* weight for each term within each document (**25 marks**).

For the querying component, you need to implement the following process steps:

- Get a user query – note that it can be set within the notebook directly. into a variable named *query*;
- Preprocess the user query (tokenisation, case-folding, stop-word removal and stemming) (**5 marks**);
- Use cosine similarity to calculate the similarity between the query and each document (**25 marks**);
- Output the list of documents as a ranked list (**10 marks**).

Note that you can use *NLTK* or any other library to help in the tokenisation and preprocessing of text. However, you **need** to implement your own **TF.IDF** weighting and **Cosine Similarity** measures.

Note also that the queries provided in the zip archive *queries-raw-texts.zip* are only to be used to get example queries. The documents that need to be processed are found in *docs-raw-texts.zip*.

3 Mark Distribution

The marks are distributed as follows:

- Indexing part:
  - Document parsing – **10 marks**;
  - Tokenisation – **5 marks**
  - Extraction of index terms – **20 marks**;
  - Weighting of index terms (term-by-document matrix) – **25 marks**
- Querying part:

  - Query processing – **5 marks**;
  - Calculate similarity between the query and each document – **25 marks**;
  - Building and outputting ranked list of relevant documents – **10 marks**.

You will need to submit the notebooks (and any other relevant files/documents) as zipped files on the VLE. Moreover, you are asked to export the Jupyter notebook as PDF and upload it to the relevant *TurnItIn* area on VLE.

You will need to demonstrate your project. The date for the demo session will be communicated later on.

4 Helpful resources

- Beginner's tutorial for Jupyter Notebook – `https://www.dataquest.io/blog/jupyter-notebook-tutorial/`
- Using NLTK – `http://www.nltk.org/book/`

5 Plagiarism

Note that this task is an individual task. While discussions between individual students are considered as healthy, the final deliverable needs to be that produced by you and not plagiarised in any way.

The submissions will be checked for plagiarism and unauthorised collusion. Disciplinary procedures will be undertaken for students found to have plagiarised or copied from each other. All sources need to be referenced accordingly.

6 Final Remarks

For any difficulties, please use the "Web Intelligence Forum"[2] on VLE. . Any issues, including technical difficulties should be identified and highlighted as early as possible to ensure timely resolution.

Good luck!!

[2] `https://www.um.edu.mt/vle/mod/forum/view.php?id=929241`