Information Extraction Final Project:
Creation of a search engine from scratch

Christophe SERVAN, PhD

1 Introduction

The main idea of this project is to set up a *search engine* in order to seek into a dataset of documents. In order to do so, you will use the CISI dataset that will be shared with you. You will use classical IR approach based on TF-IDF and BM25. In order to evaluate and compare these two approaches the performance will be measure using standart metrics : the **Precision@10**, the **Recall@10** and the **F-score@10**.

1.1 Datasets

the CISI dataset is composed of 3 files :

- **CISI.ALL** : A file of 1,460 "documents" each with a unique ID (.I), title (.T), author (.A), abstract (.W) and list of cross-references to other documents (.X). It is the dataset for training IR models when used in conjunction with the Queries (CISI.QRY).
- **CISI.QRY** : A file containing 112 queries each with a unique ID (.I) and query text (.W).
- **CISI.REL** : A file containing the mapping of query ID (column 0) to document ID (column 1). A query may map to more than one document ID. This file contains the "ground truth" that links queries to documents. Use this to test your approach.

1.2 Models

1.2.1 TF.iDF

The TF.iDF est estimated like this:

$$TF_{w,d} = \frac{\#\ occurrences\ of\ w\ in\ doc\ d}{\#\ of\ words\ in\ doc\ d}$$

$$iDF_w = \log(\frac{\#\ of\ doc\ in\ the\ collection}{\#\ of\ docs\ in\ which\ w\ occured})$$

$$TF.iDF_{w,d} = TF_{w,d} \cdot iDF_w$$

1.2.2 BM25

Pour une requête Q, contenant les mots $w_1, ..., w_n$, le score BM25 d'un document D est

$$score(Q, D) = \sum_{i=1}^{n} IDF(w_i) = \frac{TF(w_i, D) \cdot (k_1 + 1)}{TF(w_i, D) + k_1 \cdot (1 - b + b \cdot (\frac{|D|}{avgdl}))}$$

where $|D|$ is the length of the considered document (number of words) and $avgdl$ the average length of the documents in the collection.

2 Work to do

In order to realise this project, you have to **create scripts** in python to:

- load the data
- implements the Indexer, in which you will index the collection
- implements the retreiver, which will create the correspondance between the collection and the query
- implements query and data processing, like stemming, lemmatization, Named Entity Recognition or any preprocessing you will find suitable.

Finally, you have to **write a report** that contains:

- Project description
- A (short)related work section
- Bottleneck description
- Description of the solution / approach
- Result comparison / explanation / analysis
- Conclusion

You have to **share your script** in github/gitlab (in this case, please share the links). The deadline is fixed to 9th of december 2022, 23:59. **Recommendations:**

- If you encounter some issues, feel free to send me (or to Nicolas) an email, we'll answer you ASAP;
- I want you to **write a report**. This also means you have to put your name on it and to do an effort of presentation! ;-)
- The report mark is important for the final course mark (**report: 60%, code: 40%**);
- You shall work in groups (2-3 persons);

Warnings:

- If you send me your report lately, you will be penalised ;
- Plagianism equal to zero ;
- No report equal to zero ;