COMP4650/6490 Document Analysis – Semester 2 / 2023

Tutorial / Lab 2

Last updated August 2, 2023

Q1: Querying

Consider the following term-document matrix for 3 terms "quick", "brown", "fox" in a collection of 3 documents:

	quick	brown	fox
Doc1	3	0	2
Doc2	0	1	1
Doc3	0	3	6

(a) Calculate the tf-idf score of each term in each document.

(b) Now suppose that a user runs the query "quick fox". Calculate the cosine similarity between this query and each of the 3 documents, where the document and query vectors are given by the tf-idf score of each term. Which document is retrieved first?

(c) Write down the inverted index that would be created from this term-document matrix as a Python dictionary.

(d) Explain the importance of the idf component of the tf-idf score. How does the idf change the weights of rare terms and why is this useful in information retrieval?

Q2: Evaluation

Suppose that we are evaluating our IR system. For a given query, our system retrieves 10 documents, which are marked as being relevant (`R`) or irrelevant (`I`) in the following order:
`R, I, R, R, I, R, R, R, R, R`
The list is ordered left to right, so the leftmost `R` is the relevance of the first retrieved document. There are 12 relevant documents in the entire collection.

(a) Calculate the recall at 5 documents retrieved.

(b) Calculate the interpolated precision at 20% recall.

(c) Calculate the F1-score at 5 documents retrieved.

(d) Consider the task of building an IR system for a collection of legal documents (patents, court transcripts, etc) to be used by legal firms. How would you evaluate this IR system? Compare the differences in user needs between this system and a typical web search application. How would these differences influence what metrics you use to measure your system performance?

Q3: Practical exercise

In the notebook `lab2-inverted_index.ipynb` you will implement a simple indexer to construct inverted index from raw text. Work through the notebook and answer the questions in it.