Department of Computer Science
The University of Melbourne
COMP90042 WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2018)

Workshop exercises: Week 10

Discussion

1. Compare using a **term–document matrix** vs. an **inverted index** for resolving a ranked query efficiently.

2. Using the TF-IDF vector space model, using raw term frequency $f_{t,d}$, $\log \frac{N}{f_t}$ as the inverse document frequency formulation, find the ranking for the query `apple ibm`, based on calculated over the following collection. You should use cosine similarity, but for this question, we will skip the document normalisation step (for time reasons.)

	apple	ibm	lemon	sun
D_1	4	0	1	1
D_2	5	0	5	0
D_3	2	5	0	0
D_4	1	0	1	7
D_5	0	1	3	0

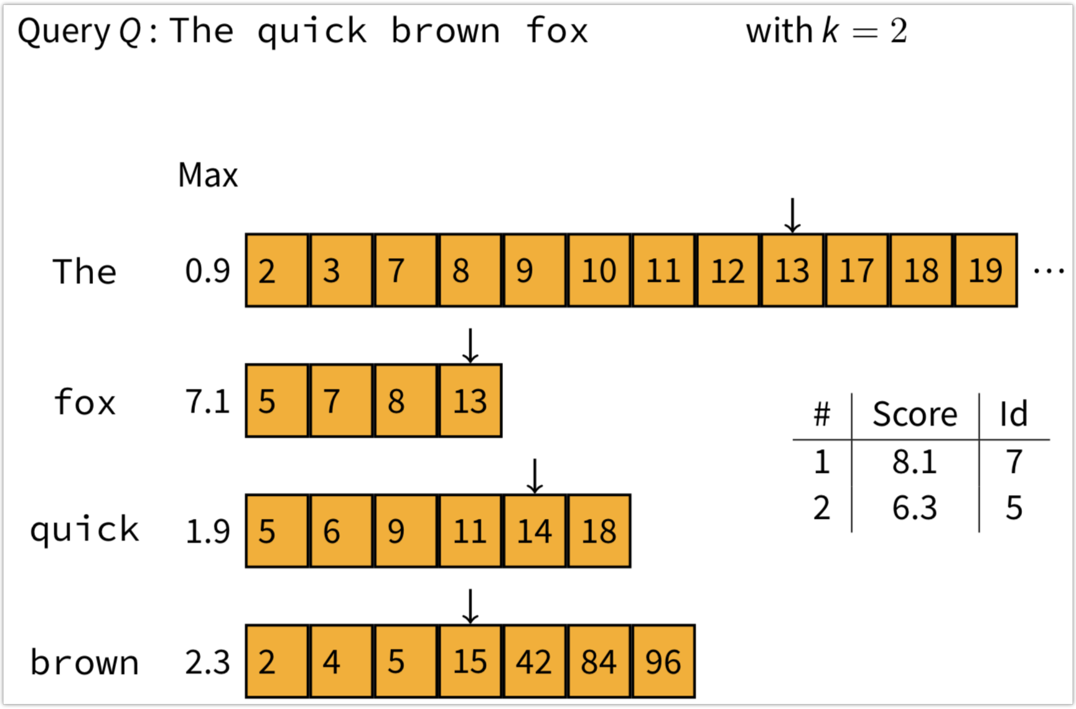
3. Recall the Okapi BM25 term weighting formula:

$$w_t = \log \frac{N - f_t + 0.5}{f_t + 0.5} \times \frac{(k_1 + 1) f_{d,t}}{k_1((1-b) + b\frac{L_d}{L_{\text{avg}}}) + f_{d,t}} \times \frac{(k_3 + 1) f_{q,t}}{k_3 + f_{q,t}}$$

   What are its parameters, and what do they signify? How do the components relate to TF (term frequency) and inverse document frequency (IDF)?

4. Data compression of a **postings list** in an inverted index can help reduce space usage of the index.
   (a) What is the intuition behind compression algorithms used for postings list compression? Why do they work?
   (b) What is **Variable Byte Compression** and how does it compress an integer?
   (c) Determine the values of integers X and Y that were encoded as the byte sequence [52,34,147,42,197] using the Variable Byte algorithm described in the lecture slides 9/10.

5. Algorithms such as WAND help speed up query processing.
   (a) What is the intuition behind WAND? What is the output produced by WAND?
   (b) What extra information is stored for each term to allow algorithms like WAND to skip evaluating documents? How is it computed? What restriction does it place on the query process?
   (c) Assume Document 13 has just been evaluated. In the setting below, what is the next document that will be evaluated?



#	Score	Id
1	8.1	7
2	6.3	5

Programming

1. Issue some queries using the small IR engine given in the iPython notebook `WSTA_N16_information_retrieval`. Read (some of) the documents that are returned: confirm that the keyword(s) is/are present, and judge whether you think these documents are relevant to your query.
2. Work on the project! :-)

Catch-up

- What is an **information retrieval engine**?
- What does it mean for a document to be **relevant** to a query?
- What is a **vector space model**? How can we find **similarity** in a vector space?

Get ahead

- What effect do the various preprocessing regimes have on the efficiency (time) and effectiveness (relevant results) of querying with the system (note: not building the index)? In particular, consider:
    1. Stemming
    2. Stopping
    3. Tokenisation (e.g. of non-alphabetic tokens)