Department of Computer Science
The University of Melbourne
COMP90042 WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2017)

Workshop exercises: Week 9

**Discussion**

1. Why is evaluating an Information Retrieval engine difficult?
   (a) What are some assumptions that we need to make, and why must we make them?
   (b) Why is recall usually impossible to calculate? Why do we (usually) not care?
2. Recall the Okapi BM25 term weighting formula:

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

   (a) What are its parameters, and what do they signify?
   (b) Using suitable default parameter settings, find the ranking for the query `apple ibm`, using the BM25 term–weighting model, as calculated over the following collection:

| | `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 | 2 | 1 | 7 |
| $D_5$ | 1 | 1 | 3 | 0 |

3. How can we use a **language model** to solve the above problem? Is this a sensible strategy? Why or why not?
   (a) What is the logic behind using a **unigram** model here? When might higher–order models be preferable, but why do we not usually use them?
   (b) Observe that the "Dirichlet smoothed" LM from the lecture slides is the same as the so–called "Jelinek–Mercer smoothed" language model from KT:

$$P(d|q) = \prod_{t \in q} \left(\frac{|d|}{|d| + \mu} \times \frac{f_{d,t}}{|d|} + \frac{\mu}{|d| + \mu} \times \frac{F_t}{F}\right)$$

**Programming**

1. 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 are **Precision** and **Recall**?
- What does **pooling** refer to, in an Information Retrieval context?
- What is a **vector space model**? How can we find **similarity** in a vector space?
- What is a **TF-IDF model**? What are some common examples of TF-IDF models?

**Get ahead**

- Adapt last week's iPython notebook to incorporate the BM25 term–weighting model. Find some queries where this system produces different results to the TF-IDF system, and explain why.