School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies (Semester 1, 2017)
Workshop exercises: Week 6

1. Given a document set made up of five documents, with the indicated term frequencies $f_{d,t}$:

DocID	`apple`	`ibm`	`lemon`	`sun`
Doc_1	4	0	0	1
Doc_2	5	0	5	0
Doc_3	2	5	0	0
Doc_4	1	2	1	7
Doc_5	1	1	3	0

calculate the document ranking for the (conjunctive) query `apple lemon`, based on the **language model** approach to IR given in the lectures, using $\mu = 1$, and then $\mu = 3$:

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

- Language models consider the probability of observing certain terms (from the query) within the document. We also "smooth" the frequencies, using the μ parameter, by adding in some extra terms according to the distribution of terms across the entire document collection. (More on this topic later in the subject.)
- The formula we use for finding the similarity of a document to a query is given above.
- There are two terms in our query: `apple` (`a`) and `lemon` (`l`). This means that there are two expressions to multiply together for each document, comprising the length of the document ($|\ d\ |$), the frequency of the term in the document ($f_{d,t}$), the frequency of the term in the entire collection (F_t), the frequency of all terms in the collection (F=38), and the smoothing parameter μ (given as 1, and we'll look at 3 later). For document 1, this looks like:

$$\begin{aligned}
S(q,d_1) &= \left(\frac{|\ d_1\ |}{|\ d_1\ |+\mu} \cdot \frac{f_{d_1,a}}{|\ d_1\ |} + \frac{\mu}{|\ d_1\ |+\mu} \cdot \frac{F_a}{F}\right) \times \left(\frac{|\ d_1\ |}{|\ d_1\ |+\mu} \cdot \frac{f_{d_1,l}}{|\ d_1\ |} + \frac{\mu}{|\ d_1\ |+\mu} \cdot \frac{F_l}{F}\right) \\
&= \left(\frac{5}{5+1} \cdot \frac{4}{5} + \frac{1}{5+1} \cdot \frac{13}{38}\right) \times \left(\frac{5}{5+1} \cdot \frac{0}{5} + \frac{1}{5+1} \cdot \frac{9}{38}\right) \\
&= \left(\frac{4}{6} + \frac{13}{228}\right) \times \left(0 + \frac{3}{76}\right) \approx 0.029
\end{aligned}$$

- So, the similarity here is about 3%. Although the formula at first looks daunting, the calculations are pretty straightforward, and we arrive at a measure of similarity straight away. Here are the calculations for the other documents:

$$\begin{aligned}
S(q,d_2) &= \left(\frac{|\ d_2\ |}{|\ d_2\ |+\mu} \cdot \frac{f_{d_2,a}}{|\ d_2\ |} + \frac{\mu}{|\ d_2\ |+\mu} \cdot \frac{F_a}{F}\right) \times \left(\frac{|\ d_2\ |}{|\ d_2\ |+\mu} \cdot \frac{f_{d_2,l}}{|\ d_2\ |} + \frac{\mu}{|\ d_2\ |+\mu} \cdot \frac{F_l}{F}\right) \\
&= \left(\frac{10}{10+1} \cdot \frac{5}{10} + \frac{1}{10+1} \cdot \frac{13}{38}\right) \times \left(\frac{10}{10+1} \cdot \frac{5}{10} + \frac{1}{10+1} \cdot \frac{9}{38}\right) \\
&= \left(\frac{5}{11} + \frac{13}{418}\right) \times \left(\frac{5}{11} + \frac{9}{418}\right) \approx 0.231 \\
S(q,d_3) &= \left(\frac{|\ d_3\ |}{|\ d_3\ |+\mu} \cdot \frac{f_{d_3,a}}{|\ d_3\ |} + \frac{\mu}{|\ d_3\ |+\mu} \cdot \frac{F_a}{F}\right) \times \left(\frac{|\ d_3\ |}{|\ d_3\ |+\mu} \cdot \frac{f_{d_3,l}}{|\ d_3\ |} + \frac{\mu}{|\ d_3\ |+\mu} \cdot \frac{F_l}{F}\right) \\
&= \left(\frac{7}{7+1} \cdot \frac{2}{7} + \frac{1}{7+1} \cdot \frac{13}{38}\right) \times \left(\frac{7}{7+1} \cdot \frac{0}{7} + \frac{1}{7+1} \cdot \frac{9}{38}\right) \\
&= \left(\frac{2}{8} + \frac{13}{304}\right) \times \left(\frac{0}{8} + \frac{9}{304}\right) \approx 0.009
\end{aligned}$$

$$
\begin{aligned}
S(q,d_4) &= (\frac{|\ d_4\ |}{|\ d_4\ |+\mu}\cdot\frac{f_{d_4,a}}{|\ d_4\ |}+\frac{\mu}{|\ d_4\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_4\ |}{|\ d_4\ |+\mu}\cdot\frac{f_{d_4,l}}{|\ d_4\ |}+\frac{\mu}{|\ d_4\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{11}{11+1}\cdot\frac{1}{11}+\frac{1}{11+1}\cdot\frac{13}{38})\times(\frac{11}{11+1}\cdot\frac{1}{11}+\frac{1}{11+1}\cdot\frac{9}{38})\\
&= (\frac{1}{12}+\frac{13}{456})\times(\frac{1}{12}+\frac{3}{152})\approx 0.012\\
S(q,d_5) &= (\frac{|\ d_5\ |}{|\ d_5\ |+\mu}\cdot\frac{f_{d_5,a}}{|\ d_5\ |}+\frac{\mu}{|\ d_5\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_5\ |}{|\ d_5\ |+\mu}\cdot\frac{f_{d_5,l}}{|\ d_5\ |}+\frac{\mu}{|\ d_5\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{5}{5+1}\cdot\frac{1}{5}+\frac{1}{5+1}\cdot\frac{13}{38})\times(\frac{5}{5+1}\cdot\frac{3}{5}+\frac{1}{5+1}\cdot\frac{9}{38})\\
&= (\frac{1}{6}+\frac{13}{228})\times(\frac{3}{6}+\frac{3}{76})\approx 0.121
\end{aligned}
$$

- For a ranked query engine, we rank the documents from most similar to least similar, according to the calculated S: document 2 is returned first, followed by 5, then 1, 4, and 3. (Note that this is the same order as the TF-IDF model from last week — but actually fewer calculations!)
- What about for $\mu = 3$?

$$
\begin{aligned}
S(q,d_1) &= (\frac{|\ d_1\ |}{|\ d_1\ |+\mu}\cdot\frac{f_{d_1,a}}{|\ d_1\ |}+\frac{\mu}{|\ d_1\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_1\ |}{|\ d_1\ |+\mu}\cdot\frac{f_{d_1,l}}{|\ d_1\ |}+\frac{\mu}{|\ d_1\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{5}{5+3}\cdot\frac{4}{5}+\frac{3}{5+3}\cdot\frac{13}{38})\times(\frac{5}{5+3}\cdot\frac{0}{5}+\frac{3}{5+3}\cdot\frac{9}{38})\\
&= (\frac{4}{8}+\frac{39}{304})\times(0+\frac{27}{304})\approx 0.056\\
S(q,d_2) &= (\frac{|\ d_2\ |}{|\ d_2\ |+\mu}\cdot\frac{f_{d_2,a}}{|\ d_2\ |}+\frac{\mu}{|\ d_2\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_2\ |}{|\ d_2\ |+\mu}\cdot\frac{f_{d_2,l}}{|\ d_2\ |}+\frac{\mu}{|\ d_2\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{10}{10+3}\cdot\frac{5}{10}+\frac{3}{10+3}\cdot\frac{13}{38})\times(\frac{10}{10+3}\cdot\frac{5}{10}+\frac{3}{10+3}\cdot\frac{9}{38})\\
&= (\frac{5}{13}+\frac{39}{494})\times(\frac{5}{13}+\frac{27}{494})\approx 0.204\\
S(q,d_3) &= (\frac{|\ d_3\ |}{|\ d_3\ |+\mu}\cdot\frac{f_{d_3,a}}{|\ d_3\ |}+\frac{\mu}{|\ d_3\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_3\ |}{|\ d_3\ |+\mu}\cdot\frac{f_{d_3,l}}{|\ d_3\ |}+\frac{\mu}{|\ d_3\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{7}{7+3}\cdot\frac{2}{7}+\frac{3}{7+3}\cdot\frac{13}{38})\times(\frac{7}{7+3}\cdot\frac{0}{7}+\frac{3}{7+3}\cdot\frac{9}{38})\\
&= (\frac{2}{10}+\frac{39}{380})\times(\frac{0}{10}+\frac{27}{380})\approx 0.022\\
S(q,d_4) &= (\frac{|\ d_4\ |}{|\ d_4\ |+\mu}\cdot\frac{f_{d_4,a}}{|\ d_4\ |}+\frac{\mu}{|\ d_4\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_4\ |}{|\ d_4\ |+\mu}\cdot\frac{f_{d_4,l}}{|\ d_4\ |}+\frac{\mu}{|\ d_4\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{11}{11+3}\cdot\frac{1}{11}+\frac{3}{11+3}\cdot\frac{13}{38})\times(\frac{11}{11+3}\cdot\frac{1}{11}+\frac{3}{11+3}\cdot\frac{9}{38})\\
&= (\frac{1}{14}+\frac{39}{532})\times(\frac{1}{14}+\frac{27}{532})\approx 0.018\\
S(q,d_5) &= (\frac{|\ d_5\ |}{|\ d_5\ |+\mu}\cdot\frac{f_{d_5,a}}{|\ d_5\ |}+\frac{\mu}{|\ d_5\ |+\mu}\cdot\frac{F_a}{F})\times(\frac{|\ d_5\ |}{|\ d_5\ |+\mu}\cdot\frac{f_{d_5,l}}{|\ d_5\ |}+\frac{\mu}{|\ d_5\ |+\mu}\cdot\frac{F_l}{F})\\
&= (\frac{5}{5+3}\cdot\frac{1}{5}+\frac{3}{5+3}\cdot\frac{13}{38})\times(\frac{5}{5+3}\cdot\frac{3}{5}+\frac{3}{5+3}\cdot\frac{9}{38})\\
&= (\frac{1}{8}+\frac{39}{304})\times(\frac{3}{8}+\frac{27}{304})\approx 0.117
\end{aligned}
$$

- This time, the document ordering is 2, followed by 5, 1, 3 and 4. Note that this time, document 3 is perceived to be more relevant than document 4 because the higher μ parameter means that the penalty for not having seen `lemon` in document 3 is smaller.

2. We ran four different systems, which each returned documents for a single query. We then judged whether each result returned was relevant (1) or not relevant (0):

Rank	1	2	3	4	5	6	7	8	9	10	11	12
System A	0	1	1	1	0	0	0	1	0	1	1	1
System B	1	0	1	0	1	0	1	0	1	0	0	0
System C	0	1	0	1	0	1	0	1	0	1	0	1
System D	1	1	1	1	1	0	0	0	0	0	0	0

(a) Find the precision of each of the four systems. What about recall?

- Precision is calculated as the fraction of returned documents that were correct; or formally, for true positives (TP; the returned documents that were actually relevant (1)) and false positives (FP; the returned documents that weren't actually relevant (0)):

$$Prec = \frac{TP}{TP + FP}$$

- System A returned 12 documents (that we are interested in here): 7 of them were relevant (1), and therefore true positives. 5 of them were irrelevant (0), and therefore false positives. So, the precision of System A is

$$\begin{aligned} Prec(A) &= \frac{TP}{TP + FP} \\ &= \frac{7}{7+5} = \frac{7}{12} \approx 58\% \end{aligned}$$

  The precision of System B is $\frac{5}{12}$, C is $\frac{6}{12}$, and D is $\frac{5}{12}$.
- Recall is calculated as the fraction of the total set of relevant documents that were actually returned, with respect to false negatives (FN; the relevant documents that weren't returned):

$$Rec = \frac{TP}{TP + FN}$$

- The problem: how do we determine FN? We would need to relevance judgements on every document in the collection, which is completely intractable (as the collection consists of billions or trillions of documents).
- One possibility is to consider the number of relevant documents returned by the system. System A returned 7 documents that were relevant, System B returned 5, System C returned 6, System D returned 5. The largest number is 7 and we can consider this as a possible TP+FN value. In that case, the recall will be: for System A $\frac{7}{7} = 100\%$, for System B $\frac{5}{7} \approx 71.4\%$, System C has $\frac{6}{7} \approx 85.7\%$, and System D has $\frac{5}{7} \approx 71.4\%$. Again, this is just one suggestion when we do not know the actual number of relevant documents.

(b) Rank the systems according to P@1, P@3, P@6, P@12.

- $P@k$ is the measurement of precision, evaluated only on the first (top) k documents returned by the system. The idea here is that, while a system probably returns many documents, a given user will only look for relevant documents among the first handful presented to them.
- For $P@1$, we consider only the first document returned by each system, and evaluate precision as usual. For System A, this is $\frac{0}{0+1} = 0$, as the first document returned is a false positive; for System B, it's $\frac{1}{1+0} = 1$, because the first document is a true positive. C and D are similarly 0 and 1; and the ranking is: B and D are greater than A and C.
- For $P@3$, we consider the top 3 documents. For System A, there are 2 true positives and 1 false positive in the top 3 documents returned, therefore $P@3$ is $\frac{2}{3}$. Systems B, C, and D score $\frac{2}{3}$, $\frac{1}{3}$, and $\frac{3}{3}$ respectively. The ranking now is D, followed by A and B, then System C.
- For $P@6$, Systems A, B, and C all have 3 true positives in the top 6 documents returned, therefore $P@6 = \frac{3}{6}$. System D is the best performer here, with $P@6 = \frac{5}{6}$.

- For $P@12$, we use the top 12 returned documents, which is all that we have judgements for — so this is equivalent to the precision we calculated in part (a). The system rank is A ($\frac{7}{12}$), then C ($\frac{6}{12}$), then B and D ($\frac{5}{12}$).
- Notice that the relative ranking of the various systems changes quite a lot depending on where we choose our $P@k$ cutoff.

(c) Make a reasonable assumption, and then find the AP score for each system.

- The reasonable assumption we need to make here is in terms of the total number of documents that are relevant to the query. Since this is a constant factor in how we calculate average precision (AP), the exact value isn't so important, as long as we apply it consistently across all of our systems. Here, we will assume there are 100 documents relevant to the query. (We could equally well assume 7 relevant documents, like in part (a).) Average precision is calculated for a system S as the arithmetic mean of all of the $P@k$ wherever a relevant document was returned. For missing relevant documents, we assume the $P@k$ is 0 (or some other suitably small number), giving the formula:

$$AP(S) = \frac{1}{N} \sum_{\{i|S(i)\text{is TP}\}} P@i$$

  where N is the total number of relevant documents, in this case 100.
- The AP of System A is:

$$\begin{aligned} AP(A) &= \frac{1}{100}[P@2 + P@3 + P@4 + P@8 + P@10 + P@11 + P@12] \\ &= \frac{1}{100}[\frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{8} + \frac{5}{10} + \frac{6}{11} + \frac{7}{12}] \\ &\approx \frac{1}{100}[4.045] \approx 0.0405 \end{aligned}$$

- For B:

$$\begin{aligned} AP(B) &= \frac{1}{100}[P@1 + P@3 + P@5 + P@7 + P@9] \\ &= \frac{1}{100}[\frac{1}{1} + \frac{2}{3} + \frac{3}{5} + \frac{4}{7} + \frac{5}{9}] \\ &\approx \frac{1}{100}[3.394] \approx 0.0339 \end{aligned}$$

- For C:

$$\begin{aligned} AP(C) &= \frac{1}{100}[P@2 + P@4 + P@6 + P@8 + P@10 + P@12] \\ &= \frac{1}{100}[\frac{1}{2} + \frac{2}{4} + \frac{3}{6} + \frac{4}{8} + \frac{5}{10} + \frac{6}{12}] \\ &= \frac{1}{100}[3] = 0.03 \end{aligned}$$

- For D:

$$\begin{aligned} AP(D) &= \frac{1}{100}[P@1 + P@2 + P@3 + P@4 + P@5] \\ &= \frac{1}{100}[\frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{4}{4} + \frac{5}{5}] \\ &= \frac{1}{100}[5] \approx 0.05 \end{aligned}$$

- Notice that AP tends to prefer systems that both have a large fraction of the documents (like recall), but also have most of the first few documents correct (like $P@k$). Notice also that the reported scores are small because all of the systems returned only a few of our assumed set of 100 relevant documents.

3. What is **pooling**, and why is it important in IR evaluation?

   - A **pool** is a collection of documents, made up of the (unique) set of top k documents returned by a number of different systems, for a single query.
   - The presumption of collecting a pool, is that each system is generally effective, and different — consequently, the pool will contain a relatively larger number of relevant results. And, more importantly, that few relevant results will not have been found by *any* of the different systems.
   - Therefore, by evaluating (by hand) the documents in the pool, we will have both: an evaluation of precision for each system; and a pretty good estimate of recall.

4. What are the four primary components of a **Web–scale Information Retrieval engine**? Briefly describe our goal in each of them.

   - **Crawling**: finding and downloading as many documents as we can from the web (hopefully all of them, although this isn't possible in practice)
   - **Parsing**: turning each document into a list of tokens (or terms), probably by removing page metadata, case folding, stemming, etc.
   - **Indexing**: building an inverted index out of all of the tokens in our downloaded document collection. (We stop worrying about the original documents at this point.)
   - **Querying**: after the previous three steps have been completed (off-line), we are ready to accept user queries (on-line), in the form of keywords, that we tokenise (in a similar manner to our document collection) and then apply our querying model (e.g. TF-IDF) based on the information in the inverted index, to come up with a document ranking
   - (Optionally) **Additional things**: change the above ranking, based on ad-hoc application of certain factors, for example, PageRank, HITS, click-through data, zones, anchor text, etc.

5. Recall the (hypothetical) method of **crawling** given in the lectures:

   (a) Would this method be *effective* at solving the problem of crawling? Why or why not?

   - Somewhat, although it depends on having a large, random set of seeds, which isn't really possible in practice. The method will miss large numbers of pages that aren't linked to from pages in the "core" of the World Wide Web, as well as all sorts of rich data encoded in databases, etc.

   (b) Would this method be *efficient* at solving the problem of crawling? Why or why not?

   - An efficient approach depends on having a good model of **web page duplication**, so that we can decide (quickly) whether a given page has already been crawled. For example, having a hash function with respect to the page's contents (although consider how large the set is, and how difficult it would be to avoid collisions!).

6. **Canonicalisation** (of text) typically comprises **tokenisation** and **normalisation**. What are these generally accepted as referring to?
   (Note the terminology is not used consistently in the literature; for example "tokenisation" occasionally refers to all three ideas.)

   - Canonicalisation, here, means having a single representation of a text which we can use to sensibly compare one text with another (in our case, a document on the Web, and a query).
   - Tokenisation, here, means decomposing the larger document into smaller information–bearing units ("tokens") than we can compare against (the keywords present in) our query.
   - Normalisation, here, means transforming a token into a form which is generally representative of other instances of the same idea (for example, by correcting spelling).

   (a) What are some issues that arise when canonicalising text written in English?

   - From the lectures:

- Stopwords
- Stemming
- Date/number formatting
- Dialect variation
- Spelling errors
- Hyphenated tokens
- Compound words
- The genitive *'s*

- Note that English is **easy** compared to many languages!

(b) (EXTENSION) What are some issues that might arise when canonicaling text written in other languages?