School of Computing and Information Systems
The University of Melbourne
COMP90049 Knowledge Technologies (Semester 2, 2018)
Workshop sample solutions: Week 5

1. What is the difference between "data retrieval" and "information retrieval"? Why is the latter a knowledge task, but the former is not?

    - The main difference here is the existence of people — users. Because people are wildly divergent, the notion of a relevant result in information retrieval depends on contextualising the data to the particular user (which may be very difficult, because we have an imperfect model of the user and indeed the user has an imperfect model of their needs!). Whereas with data retrieval, there is a particular unit of data (bitstream) that we need to access in memory or on a hard drive, and there is generally no ambiguity.

2. [**EXTENSION**] How many books are there in an average library? How many words are there in an average library? How many documents are there on the World Wide Web? How many words?

    - These aren't straightforward questions to answer. But, to an order of magnitude, a small city library might have about 10K books; a larger one, maybe 30K. I might estimate the word count of a typical book to be about 50K (many are longer; many are shorter; there are varying definitions of "word"), which would situate a library as carrying roughly 1G words. The US Library of Congress catalogues about 2.3M books, so perhaps 100G words.
    - As of 2008, Google claimed to index 1T unique urls (`http://googleblog.blogspot.com.au/2008/07/we-knew-web-was-big.html`); by 2012, this had supposedly risen to 30T (`http://www.google.com/insidesearch/howsearchworks/thestory/`). But maybe take all of this with a grain of salt! `:-)` Estimating the number of words on the Web is even harder — Google tells me that the mean document size is about 400KB, but much of that isn't going to be text. I might ballpark about 1000 words (roughly 6KB of the 400KB) per document, which might be upwards of 10000T words (or more!, but probably less)!

3. Identify some different types of "informational needs."

    - Requests for informations, e.g. "global warming"
    - Factoid questions, e.g. "what is the melting point of lead?"
    - Topic tracking, e.g. "what is the history of this news story?"
    - Navigational, e.g. "University of Melbourne home page"
    - Service or transactional, e.g. "Mac powerbook"
    - Geospatial, e.g. "Carlton restaurant"

    This isn't an exhaustive list. Nor is it non-overlapping: for example, any query can be construed as being navigational in nature (as the user is likely to click through to a relevant document), and most are informational as well.
    The three categories that are typically given are *informational, navigational, transactional*; but, again, these are somewhat arbitrary themselves.

4. Identify some differences between Boolean querying and ranked querying.

    - Boolean: documents match if they contain the terms (and don't contain the `NOT` terms; i.e. the Boolean formula evaluates to `TRUE`); matching is Yes/No; repeatable, auditable, controllable; queries allow expression of complex concepts
    - Ranked: based on evidence that the document is on the same topic as the query; matching is gradiated (to come up with a ranking!); different models give different results; queries are easy to write and results are easy to read for non-specialists

5. Identify the two (sometimes three) components of “TF-IDF” models. Indicate the rationale behind them as in, why would they contribute to a “better” result set?

   - More weight is given to documents where the query terms appear many times (TF)
   - Less weight is given to terms that appear in many documents (IDF)
   - Less weight is given to documents that have many terms (sometimes)

6. 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) queries: (a) `apple` and (b) `apple lemon`, based on the following TF-IDF term weighting model:

$$
\begin{aligned}
w_{d,t} &= \begin{cases} 1+\log_2 f_{d,t} & \text{if } f_{d,t} > 0 \\ 0 & \text{otherwise} \end{cases} \\
w_{q,t} &= \begin{cases} \log(1+\frac{N}{f_t}) & \text{if } f_{q,t} > 0 \\ 0 & \textit{otherwise} \end{cases}
\end{aligned}
$$

   - We’ve been given a weighting model, TF-IDF; implicitly, we’re going to use the cosine similarity to compare documents and queries.

   - First, we need to calculate the weights of terms in the documents; for example, the weight of `apple` in document 1 is $1 + \log_2(4)$, because we have seen 4 instances of the term `apple` ($f_{1,a} = 4 > 0$):

$$
\begin{aligned}
w_{1,a} &= 1+\log_2(4) = 3 \\
w_{1,i} &= 0 \\
w_{1,l} &= 0 \\
w_{1,s} &= 1+\log_2(1) = 1
\end{aligned}
$$

$$
\begin{aligned}
w_{2,a} &= 1+\log_2(5) \approx 3.32 \\
w_{2,i} &= 0 \\
w_{2,l} &= 1+\log_2(5) \approx 3.32 \\
w_{2,s} &= 0
\end{aligned}
$$

$$
\begin{aligned}
w_{3,a} &= 1+\log_2(2) = 2 \\
w_{3,i} &= 1+\log_2(5) \approx 3.32 \\
w_{3,l} &= 0 \\
w_{3,s} &= 0
\end{aligned}
$$

$$
\begin{aligned}
w_{4,a} &= 1 + \log_2(1) = 1 \\
w_{4,i} &= 1 + \log_2(2) = 2 \\
w_{4,l} &= 1 + \log_2(1) = 1 \\
w_{4,s} &= 1 + \log_2(7) \approx 3.81
\end{aligned}
$$

$$
\begin{aligned}
w_{5,a} &= 1 + \log_2(1) = 1 \\
w_{5,i} &= 1 + \log_2(1) = 1 \\
w_{5,l} &= 1 + \log_2(3) \approx 2.58 \\
w_{5,s} &= 0
\end{aligned}
$$

- Next, we need the weights for the query vector (let's start with `apple`). This is based on the inverse document frequency (IDF), which in this case is the inverse of the proportion of documents in which the term appears (f_t) out of the total number of documents ($N = 5$). We could also do this on a per-term basis, but we would need to re-visit our notion of what "rare" means in this context.

$$
\begin{aligned}
w_{q,a} &= \log_2(1 + \frac{5}{5}) = 1 \\
w_{q,i} &= 0 \\
w_{q,l} &= 0 \\
w_{q,s} &= 0
\end{aligned}
$$

- Let's summarise. At this point, our representation of the documents (and query) are the following 4-D vectors:

$$
\begin{aligned}
\text{Doc}_1 &: \langle 3, 0, 0, 1 \rangle \\
\text{Doc}_2 &: \langle 3.32, 0, 3.32, 0 \rangle \\
\text{Doc}_3 &: \langle 2, 3.32, 0, 0 \rangle \\
\text{Doc}_4 &: \langle 1, 2, 1, 3.81 \rangle \\
\text{Doc}_5 &: \langle 1, 1, 2.58, 0 \rangle \\
q &: \langle 1, 0, 0, 0 \rangle
\end{aligned}
$$

- To use the vector space model, we calculate the cosine similarity based on the inner product (dot product), normalised by the vector norms[1] (length; this also accounts for the document length component in our model):

$$
cos(A, B) = \frac{A \cdot B}{\mid A \mid \quad \mid B \mid}
$$

- So, the scores for our 5 documents:

$$
\begin{aligned}
cos(\text{Doc}_1, q) &= \frac{\text{Doc}_1 \cdot q}{\mid \text{Doc}_1 \mid \quad \mid q \mid} \\
&= \frac{\langle 3, 0, 0, 1 \rangle \cdot \langle 1, 0, 0, 0 \rangle}{\mid \langle 3, 0, 0, 1 \rangle \mid \quad \mid \langle 1, 0, 0, 0 \rangle \mid} \\
&= \frac{3 * 1 + 0 * 0 + 0 * 0 + 1 * 0}{\sqrt{3^2 + 0^2 + 0^2 + 1^2}\sqrt{1^2 + 0^2 + 0^2 + 0^2}} \\
&= \frac{3}{\sqrt{10}\sqrt{1}} \approx 0.95
\end{aligned}
$$

[1] Note that we can pre-calculate the (vector) length of our documents as soon as we've seen them (and not at query-time, when the user is waiting). Note also that the query length is irrelevant, because it's the same factor for every document, and all we really care about is the document ordering, not the actual scores.

$$
\begin{aligned}
cos(\mathrm{Doc}_2, q) &= \frac{\langle 3.32, 0, 3.32, 0\rangle \cdot \langle 1, 0, 0, 0\rangle}{\mid \langle 3.32, 0, 3.32, 0\rangle \mid \quad \mid \langle 1, 0, 0, 0\rangle \mid} \\
&= \frac{3.32 * 1 + 0 * 0 + 3.32 * 0 + 0 * 0}{\sqrt{3.32^2 + 0^2 + 3.32^2 + 0^2}\sqrt{1^2 + 0^2 + 0^2 + 0^2}} \\
&\approx \frac{3.32}{\sqrt{22}\sqrt{1}} \approx 0.71
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_3, q) &= \frac{\langle 2, 3.32, 0, 0\rangle \cdot \langle 1, 0, 0, 0\rangle}{\mid \langle 2, 3.32, 0, 0\rangle \mid \quad \mid \langle 1, 0, 0, 0\rangle \mid} \\
&\approx \frac{2}{\sqrt{15}\sqrt{1}} \approx 0.52
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_4, q) &= \frac{\langle 1, 2, 1, 3.81\rangle \cdot \langle 1, 0, 0, 0\rangle}{\mid \langle 1, 2, 1, 3.81\rangle \mid \quad \mid \langle 1, 0, 0, 0\rangle \mid} \\
&\approx \frac{1}{\sqrt{20.5}\sqrt{1}} \approx 0.22
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_5, q) &= \frac{\langle 1, 1, 2.58, 0\rangle \cdot \langle 1, 0, 0, 0\rangle}{\mid \langle 1, 1, 2.58, 0\rangle \mid \quad \mid \langle 1, 0, 0, 0\rangle \mid} \\
&\approx \frac{1}{\sqrt{8.7}\sqrt{1}} \approx 0.34
\end{aligned}
$$

- The best values for the cosine similarity measure are those closest to 1, because then the angle is small (close to 0), which means that the vectors point in the same direction, which means that the (weighted) distribution of terms in the document is similar to the distribution of terms in the query (which is what we want!)
- In this case, the best document is 1 (with a score of 0.95), and then 2, 3, 5 and 4.
- The query `apple lemon` is similar to `apple`, except that we have two non-zero terms in the query vector; this time, we observe that `lemon` occurs in $\frac{3}{5}$ of the document set:

$$
\begin{aligned}
w_{q,a} &= \log_2(1 + \frac{5}{5}) = 1 \\
w_{q,i} &= 0 \\
w_{q,l} &= \log_2(1 + \frac{5}{3}) \approx 1.42 \\
w_{q,s} &= 0
\end{aligned}
$$

- At this point, I might mention that any terms missing from the query can be safely ignored in the calculation of the dot product and the query length (they still need to be considered for the document length, but we probably calculated that before we saw the query anyway).
- We don't need to change our document model explicitly, but implicitly I'm only going to be concerned with the `apple` and `lemon` dimensions for the dot product — the document lengths are still the same as they were above, so I've still written the full vector in the denominators below. Now we just repeat the calculations with the new query vector $q : \langle 1, 1.42\rangle$:

$$
\begin{aligned}
cos(\mathrm{Doc}_1, q) &= \frac{\mathrm{Doc}_1 \cdot q}{\mid \mathrm{Doc}_1 \mid \quad \mid q \mid} \\
&= \frac{\langle 3, 0\rangle \cdot \langle 1, 1.42\rangle}{\mid \langle 3, 0, 0, 1\rangle \mid \quad \mid \langle 1, 1.42\rangle \mid} \\
&= \frac{3 * 1 + 0 * 1.42}{\sqrt{3^2 + 0^2 + 0^2 + 1^2}\sqrt{1^2 + 1.42^2}} \\
&\approx \frac{3}{\sqrt{10}\sqrt{3}} \approx 0.55
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_2, q) &= \frac{\langle 3.32, 3.32\rangle \cdot \langle 1, 1.42\rangle}{\mid \langle 3.32, 0, 3.32, 0\rangle \mid \quad \mid \langle 1, 1.42\rangle \mid} \\
&= \frac{3.32 * 1 + 3.32 * 1.42}{\sqrt{3.32^2 + 0^2 + 3.32^2 + 0^2}\sqrt{1^2 + 1.42^2}} \\
&\approx \frac{8}{\sqrt{22}\sqrt{3}} \approx 0.98
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_3, q) &= \frac{\langle 2, 0\rangle \cdot \langle 1, 1.42\rangle}{\mid \langle 2, 3.32, 0, 0\rangle \mid \quad \mid \langle 1, 1.42\rangle \mid} \\
&\approx \frac{2}{\sqrt{15}\sqrt{3}} \approx 0.30
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_4, q) &= \frac{\langle 1, 1\rangle \cdot \langle 1, 1.42\rangle}{\mid \langle 1, 2, 1, 3.81\rangle \mid \quad \mid \langle 1, 1.42\rangle \mid} \\
&\approx \frac{2.42}{\sqrt{20.5}\sqrt{3}} \approx 0.31
\end{aligned}
$$

$$
\begin{aligned}
cos(\mathrm{Doc}_5, q) &= \frac{\langle 1, 2.58\rangle \cdot \langle 1, 1.42\rangle}{\mid \langle 1, 1, 2.58, 0\rangle \mid \quad \mid \langle 1, 1.42\rangle \mid} \\
&\approx \frac{4.7}{\sqrt{8.7}\sqrt{3}} \approx 0.91
\end{aligned}
$$

- This time, document 2 is best (0.98), followed closely by 5, then 1, 4 and 3.