CA4009: Search Technologies
Laboratory Session 1
3rd November 2016

Jordan Healy - 13379226
Tríona Barrow - 11319851

- *Document Collection Statistics*

From clicking on “View Collection Stats” we can see we have a list of up to 100 terms. Some of these have been stemmed down, such as “countri” and “financi”, which has increased the collection and document frequencies. We can also see a number of individual numbers that are showing up. Some of these terms will be useful, however certain terms, such as single or double digit numbers, would be generic and would not apply to specific sections of the documents. So even though their frequencies are pretty high, they won’t relate to any specific information.

We also notice that one that has popped up frequently is “_an” - this appears the be an error when the documents were uploaded and/or indexed. All other stop words have been removed, so this one seems to have slipped through due to the underscore. With some stop words also, such as “dai” - it is difficult to work out what word it is linked to due to overstemming, so this could relate to multiple words, such as “daisy” and “daily”

- *Interactive Learning Using Lucene*

We started with the Okapi BM25 ranking function, and searched for “finance 1992”, setting k as 1.2 and b as 0.75, as recommended. From playing with the b value, we can change the rate of normalisation for the documents retrieved, which can impact on the amount of occurrences for our search terms. We also moved around k, seeing that some of the results became less obviously related to the search terms, as it searched for the highest number of occurrences for any of the search terms, dropping “finance” altogether in the first result when we moved it to 10. We can also see from the snippets produced that it is searching both terms independently, so both terms may not appear in the same snippets/document depending on how large k is.

Next we tried to use tf-idf, from a first glance we can see that the documents and snippets returned seem to be shorter. It appears they are ranked using Luhn’s understanding of semantic relevance, as opposed to the frequency of occurrences - in that they’re ranked by the least amount of words between the words in the search term. Both words are still being searched independently, and this impacts the length of the documents/snippets as we move further down the results page. The results from tf-idf seem to strongly resemble BM25 when k is set to 1.2 and b is set to 1, giving very similar results.

- *Coding Exercise*

Part 1

For the first part of the program we updated the `showTerms()` method to store the values, and used a `Comparator` and `Collections.sort()` to sort the key-value pairs by the value. This means they were sorted by the frequency of the words (value), as opposed to alphabetically by the tokens (key). We did this two values at a time, then used an `Iterator` to help control the number of lines output. The only change to main that we made was to store the `args[0]` value - k number of terms.

Output:

```
healyj36@lg25-30:~/Desktop/WordCounter_JAVA/src/wordcounter $
java WordCounter 10
the: 37
information: 17
of: 17
in: 14
retrieval: 13
to: 13
a: 12
are: 8
by: 8
for: 8
```

Part 2

For this part, we only modified `getTerms()`, and we added a `toLowerCase()` method when adding to the `terms List`. This was to ensure that the frequencies were accurate in `showTerms()`.

Output:

```
healyj36@lg25-30:~/Desktop/WordCounter_JAVA/src/wordcounter $
java WordCounter 10
the: 43
information: 18
in: 17
of: 17
retrieval: 14
to: 13
a: 12
by: 9
are: 8
for: 8
```

Part 3

For this section we used `PorterStemmer.java` by calling an instance of it, then using it in `getTerms()` passing the current `StringToken` to it. We ensured it was lowercase first then we added it to our `PorterStemmer` instance, stemmed it, converted back to a string and added it to the `terms List` (similar to `main` in `PorterStemmer`).

Output

```
healyj36@lg25-30:~/Desktop/WordCounter_JAVA/src/wordcounter $
java WordCounter 10
the: 43
inform: 18
in: 17
of: 17
retriev: 14
to: 13
a: 12
search: 10
system: 10
by: 9
```