Praktikum 4

DRAFT

In this lab we will be using the TERRIER information retrieval platform to explore text indexing. We will use the Cranfield document collection, consisting of a series of English-language abstracts for scientific articles, mostly concerning physics, engineering or aeronautics.

Part I

We will use a Java-based interface to the TERRIER platform, which includes the Cranfield collection. To use this collection, select **CF** as the source.

Exercise 1

a) Build a term index by clicking the green arrow. This should take a few seconds. Next to each term we can see the Document Frequency and the Collection Frequency. For example, the line `uniform,term143 df=123 cf=180` means that the term `uniform` is seen in 123 different documents, and 180 times within all of these documents. How many unique terms are there in this collection?

b) Using the **Indexing Components** section, we can add or remove preprocessing features. Try enabling the English stopword list and re-building the index. How many terms are in the stopword list?

c) Next, turn on English-language stemming and re-build the index. What happens to the number of unique terms and why?

Exercise 2

a) Enter and run a query, for example `fluid density`, using the box provided. How many documents contain these terms?

b) Check the highest-scoring document, using the document ID provided, in `/collections/cranfield/docs`. Is it relevant to the query?

c) Next, try a natural-language query, such as `what is the impact of jet noise on structural integrity`. Are the top-ranked documents relevant?

d) Enter the same query as part c), but with BM25 weighting instead of the default TFIDF. Do the most relevant documents change?

Exercise 3

a) A query Q is comprised of query terms $q_1, q_2, \ldots, q_n$. If a document d is returned by the TERRIER system, which of the following applies:

i) q_1 **and** q_2 **and** $q_3 \ldots$ **and** q_n are contained in d.

ii) q_1 **or** q_2 **or** $q_3 \ldots$ **or** q_n are contained in d.

What queries would you run to demonstrate your answer?

b) In the case of both i) and ii), would adding extra query terms increase precision or recall?

c) Let D_{q_i} be the set of documents containing q_i. Which of the following represents the set of documents returned by TERRIER for a query Q?

i) $D_{q_1} \cup D_{q_2} \cup \ldots \cup D_{q_n}$

ii) $D_{q_1} \cap D_{q_2} \cap \ldots \cap D_{q_n}$

d) What would be the advantages/disadvantages of both i) and ii), from a user's perspective?

Part II

In this part, enable the English-language stemmer, but disable stopword removal, unless instructed otherwise. Remember to re-build the index where required.

Exercise 4

a) Run two queries for the term `olive`: one with stemming enabled, and one without stemming. Do the results differ, and why?

b) Check each document that is returned – are there any documents in this collection that concern olives?

c) Therefore, do these results represent a successful or unsuccessful application of stemming?

d) Next, query the collection for the nonsense word `olivation`, that should be present in no documents. Does the stemmer help or hinder retrieval in this case?

e) Let $s(t)$ be a function which produces a stemmed version of term t, and let $m(t)$ map a term t to its semantic meaning. Which of the following statements are true?

$$t_1 = t_2 \implies s(t_1) = s(t_2) \tag{1}$$
$$s(t_1) = s(t_2) \implies t_1 = t_2 \tag{2}$$
$$s(t_1) = s(t_2) \implies m(t_1) = m(t_2) \tag{3}$$
$$m(t_1) = m(t_2) \implies s(t_1) = s(t_2) \tag{4}$$
$$t_1 = t_2 \implies m(t_1) = m(t_2) \tag{5}$$
$$m(t_1) = m(t_2) \implies t_1 = t_2 \tag{6}$$

If a statement is false, find a counter-example to demonstrate this.

Exercise 5

Suppose we have a multi-term query, but we need to give special priority to one of the terms. For instance, we wish to know about `rocket fuel ignition`, but want to make sure we only see documents about rockets, not just any kind of fuel.

i) If we enter an important word twice in a query, would it have double the weight in the final rankings?

ii) Suppose this doesn't work: can we use the stemmer to trick the system into accepting the same word twice? Try this with the words `rocket` and `rockets`, which both stem to the same word.

Weighting schemes like TFIDF assign value to terms in a corpus based on their 'importance', but each individual user may find one word more important than another, regardless of how often it appears. Consider the following two augmented weighting schemes, where $W(q_i)$ is the original weight of a term:

i) A query now takes the form $Q = (w_1, q_1), (w_2, q_2), \ldots, (w_n, q_n)$, where each term is accompanied by a weight provided by the user. The new weighting scheme is $W'(q_i) = W(q_i) \cdot w_i$. For example:

```
1,vapour 4,trails 4,jet 2,engine -1,propeller
```

This gives greater emphasis to the terms `trails` and `jet`, and negative weighting to `propeller`.

ii) $W'(q_i) = W(q_i) \cdot (n - i)$, where the first term has the most emphasis, and each subsequent term is less important than the previous one. For example:

```
jet trails engine vapour
```

Would emphasise `jet` and `trails` over the other terms.

What are the advantages and disadvantages of each of the two schemes?

Exercise 6

a) Try running the queries `jet` and `the jet`, with TFIDF enabled, but no stopword removal. Do the top-ranked documents change at all?

b) In a TFIDF weighting scheme, the term `the` will be weighted so low that it is practically negligible. Because of this, is it necessary to perform stopword removal at all when TFIDF is enabled? Why or why not?

c) Run the query `what is the optimum if the flow is drag free` and note the top-ranked document.

d) Next, try running the same query, but without any stopwords, i.e. `optimum flow drag free`. Does the top-ranked document change?

e) What if all the non-stopwords were removed, i.e. `what is the if` – does the top ranked document match the one from part (c) or part (d), or neither?

f) Now run the queries from part (a) again, but this time enable BM25 weighting instead of TFIDF. Do the top-ranked documents change when `the` is removed? Is this the same result as TFIDF? Why or why not?

g) Based on these results, would you still give the same answer to part (b)?

Exercise 7

a) Look at the index for the Cranfield collection. How many documents contain the term `zurich`? What about `zero`?

b) How many documents contain either of these terms?

c) How many contain both of these terms together?

d) What about documents containing both `wave` and `wavelength`?

e) How many documents contain *all* of the terms `air`, `aircraft` and `wing`? How about `air` and `wing`, but not `aircraft`?

Part III

Exercise 8

Suppose that the number of times a term t appears in document d is represented by $C_{t,d}$. We can calculate the term frequency, $\text{TF}(t, d)$, in a number of different ways, for instance:

i) Term count:

$$\text{TF}(t, d) = C_{t,d}$$

ii) Binary:

$$\text{TF}(t, d) = \begin{cases} 0 & \text{if } t \notin d, \\ 1 & \text{if } t \in d. \end{cases}$$

iii) Document length normalised:

$$\text{TF}(t, d) = \frac{C_{t,d}}{\sum_{t' \in d} C_{t',d}}$$

By querying the collection for the term `unmodulated` (and using the index), can you find out which of the three TF functions is used by this system?
Suppose we use a stopword remover in conjunction with method i), what would be the effect on $\text{TFIDF}(t, d)$? What about using methods ii) and iii)?

Exercise 9

a) Run a query for the term `vapour`, with stemming enabled. Note the ID of the top ranked document.

b) Next, run a query for `wing air aircraft` and again note the top ranked document ID.

c) Finally, run a query for `vapour wing air aircraft`. Is the top ranked document the same as part a), b), or something else?

d) Why do you think this is, and is this an example of the system performing correctly?

e) Which of the weighting schemes shown in Exercise 8 would give the best results on this query?

f) Which of those weighting schemes is preferable for single-term queries?

g) If the previous two answers were different, can you think of a way to achieve good performance in both single- and multi-term queries?

Exercise 10

a) Let $\text{TFIDF}(t, d_n)$ be the TFIDF weight of the term t in document number n, and $\text{IDF}(t)$ be the IDF value for term t.

   i) Why does the IDF function not require a document id?

   ii) What is the value of $\text{TFIDF}(\texttt{pod}, d_{725})$?

   iii) What is the value of $\text{TFIDF}(\texttt{rib}, d_{640})$?

   iv) How about $\text{IDF}(\texttt{rib})$ and $\text{IDF}(\texttt{pod})$?

b) How would using a stopword list affect the value of $\text{IDF}(t)$ for any given t?

c) Suppose you were to add a new document to the collection, consisting of only stopwords - what would happen (if anything) to $\text{IDF}(t)$?

Exercise 11

a) In this system, is stopword filtering applied before or after stemming? How would you show this?

b) Give an example of a situation where it would be disadvantageous to perform stopword removal after stemming.

c) Suppose we calculate the TFIDF values before removing any stopwords. Which of the methods in Exercise 8 would give a change in weight for a term t?