13 10 2016 **SEWN 2016 labsheet 2 (assessed coursework)**
Ranking content with TF-IDF

This labsheet builds upon labsheet 1. In that labsheet you were required to syndicate dynamic content from a web service. In labsheet 2, we require you to rank the content sourced from your web service using TF-IDF, a popular text weighting metric in information retrieval.

Part 1: TF-IDF

Find out about term frequency-inverse document frequency (TF-IDF), and how it can be calculated. According to http://www.tfidf.com/ TF-IDF is "a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus." TF-IDF is frequently used in recommendation and ranking applications.

Examples of TF-IDF are given at:

- http://www.tfidf.com/
- https://en.wikipedia.org/wiki/Tf-idf

Other examples can be found on-line.

Part 2: Gathering data

Using your web service from labsheet 1, we require you to gather data from it until you have a corpus of between 50-100 documents to give a word count of approximately 2,500 words. See Part 5 below for a description of the term *document* in relation to an RSS feed as an example of data gathered.

Part 3: Ranking

We want you to apply TF-IDF to rank the keywords found in your document corpus as a whole and extract from this list the top 25 keywords and their rankings. In doing this, you are free to employ on-line tools/programs or to write code of your own but do not be concerned with excessive details in your calculations.

The top 25 keywords mined should all be proper nouns, e.g. names of entities, events, locations and so on. Examples might include *Theresa May*, *Brexit*, *Barack Obama*, *London*, *Parliament* and so on. You can restrict your keywords to unigrams, i.e. single words, but if you do so a name like *Barack Obama* will become two words, i.e. *Barack* and *Obama*. Alternatively, you can try to use bigram or trigram keywords of two or three words respectively.

Do not include stop words as keywords. Stop words are common words such as conjunctions, determiners or prepositions which are not inherently meaningful and are often filtered in information retrieval. Examples of stop words include *and*, *at*, *certain*, *is*, *meanwhile*, *on*, *the,* and *which*. You can find out more about stop words at:

https://en.wikipedia.org/wiki/Stop_words

If you think that you do not have sufficient data from your data source, martin@dcs.bbk.ac.uk can provide data from older RSS feeds.

Part 4: Visualisation

A visualisation(s) of the TF-IDF rankings of the top 25 keywords in your document corpus. Simple *x*, *y* or pie charts displaying the keywords and their rankings are sufficient but extra credit will be given for unusual or interesting visualisation types, e.g. use of word (tag) clouds, tree-maps, bubble charts or other such types.

Part 5: Building a document corpus

For Part 2 above we require that you gather data from your data source as a series of documents because TF-IDF calculates the relevance, i.e. ranking, of each word in a *collection of documents*. An RSS feed can be thought of as a series of documents.

How can we do this? Stated simply, RSS is a dialect of XML and an RSS feed is made up of a `<channel>` element which describes the feed, and a set of one or more `<item>` elements, each of which contains a story.

In the mock RSS feed below, there are three `<item>` elements:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
    <channel>
        <title>UFP News</title>
        <link>http://www.ufp.com/</link>
        <description>The latest news from the United Federation of Planets.</description>
        <item>
            <title>Klingon ambassador storms out of council</title>
            <description>The Klingon ambassador stormed out of the council chamber earlier
                         today when the Romulans announced economic sanctions.</description>
            <pubDate>Fri, 02 Jan 2380 20:07:36 GMT</pubDate>
        </item>
            <title>Starfleet Command goes on strike</title>
            <description>Starfleet Command went on strike at Stardate 456789.3 in support
                         of the Klingons and their stance against the Romulan economic
                         sanctions.</description>
            <pubDate>Fri, 02 Jan 2380 20:07:38 GMT</pubDate>
        </item>
        <item>
             <title>Venusian intervention on Mars?</title>
             <description>Is Venusian intervention on Mars inevitable? That is the
                          question being asked by pundits today. Many expect the president
                          of Mars to declare martial law.</description>
             <pubDate>Fri, 02 Jan 2380 20:07:40 GMT</pubDate>
        </item>
    </channel>
</rss>
```

Each `<item>` is a story and corresponds to a *document.* We want you to build your documents by either:

a. Extracting and concatenating the contents of the `<title>` and `<description>`, elements from each `<item>`, e.g. where a document would contain the text:

   Klingon ambassador storms out of council The Klingon ambassador stormed out of the council chamber earlier today when the Romulans announced economic sanctions.

b. Extracting the `<description>` element text of each `<item>`, e.g. where a document would contain text:

   Starfleet Command went on strike at Stardate 456789.3 in support of the Klingons and their stance against the Romulan economic sanctions.

Each document then forms one of the 50-100 documents to give a word count of approximately 2,500 words required of Part 2.

Document `sewn_2016_labsheet_2_addendum.pdf` lists two examples of TF-IDF rankings, one of which is for the text corpus above.

What to hand in:
Submit a single `.zip` file via the **Labsheet 2 Ranking content with TF-IDF** drop box in Moodle containing:

a. A link to your web service.
b. A short report (of no more than two A4 pages) in `.doc(x)` or `.pdf` format. The report should include the TF-IDF rankings of the top 25 keywords from your document corpus for Part 3, together with any thoughts you may have concerning these, and your visualisation(s) of the rankings for Part 4.
c. Any program code, or equivalent implementation, should be supplied together with any special instructions for compiling and running your solution (if there are several code files, please include the folder structure in your `.zip` file described below). Depending upon your solution, you may be required to demonstrate it in one of the PC laboratories at the college.
d. The data gathered from your data source in either of the following formats: (1) each document should be supplied in a separate text file named `doc_x.txt` where *x* refers to the document number, or (2) by hardcoding the data into your program where each document forms an element of a data structure. **The second option is preferred.**
e. Any references to books or on-line material for this labsheet should be included in your report.

Submission deadline: 27 10 2016

Late assignments: No extensions are available as for this labsheet and any late submissions will be graded as per the guidelines of the relevant course being studied.