# Portfolio 2

**The Guardian API**

The British newspaper The Guardian was first released in 1821, and is now published in the UK, the US and in Australia. The Guardian Open Platform gives free access to almost two million items of content, access to full text, and has a very user friendly and developed documentation. This portfolio is working with the Guardian API, and the results are from all branches of the guardian, and all sections.

**Importing the text**

The text is imported by using a script found on github[[1]](#footnote-1) that collects all content between to dates, set as parameters for the API request. It continues to make a list of texts. This is the corpus used in the analysis.

Looking at the data

To get a feel of the text some simple functions is called. To find out the size of the corpus, the number of texts is called. To have a closer look, the random.sample function is called on full text. This gives a view of two full texts articles. The number of words is also counted by calling the word tokenize function in the nltk library, and then making a list of all words, and the random function called again. This is all to check the import and see if the corpus makes sense.

Number of words and sentences is used to calculate the average number of words and sentences per article. This gives a sense of the types of texts the corpus is comprised of. If these are articles, they are fairly long reads.

Term document matrix

Using a term document matrix allows us to make a matrix with an index for texts, and index for words, and a grid with values for every word used in the texts. The stop words are removed in this step, as well as normalizing the vocabulary with a token pattern.

SKlearns count vectorizer is imported, and the count vectorizer is fitted to the vocabulary of the texts, and then the transform function is called to encode the texts to the vector.

This matrix has a very hi sparsity which means most of its values are zero. The matrix allows us to find the top words used after cleaning the data.

Applying TF-IDF weighting

The term frequency-inverse document frequency matrix gives the weight of the words or the relevance. This means that doing the same calculation on the new TF-IDF matrix, the results are much more rare words with more meaning, as the new matrix not just counts the frequency words, but sets it against how common they are in the text corpus. This gives us the relevance for each document.

Making a query

The TF-IDF matrix makes it possible to find the documents in the text corpus which has the best correlation with specific terms. Using cosine similarity it can give an overview of which documents are most relevant to look at, in order to make a topic modelling analysis. Here the top ten most relevant documents is shown.

**Topic modelling**

Topic modeling works with a bag-of-words approach, and that a topic can be shown relation between the meaning of each word as a cluster (Ignatow, G. & Mihalcea 2018). It is widely used to discover correlations between discourses in journalism and politics, and in this assignment the goal was to select a corpus relevant to a query, and perform topic modelling of this corpus. As this was not within the competences of the author, the topic modelling is performs on the full corpus. Finding a solution to this problem will be the next step, working with the Guardian API.

**Literature**

Ignatow, G. & Mihalcea, R. (2018) An Introduction to Text Mining : Research Design, Data Collection, and Analysis.Los Angeles, CA. Sage

Greene, D. and Cross, J.P. 2017. Exploring the political agenda of the european parliament using a dynamic topic modeling approach. Political Analysis 25(01), pp. 77–94.

Code

1. <https://gist.github.com/fsluis/1fef039e7a4065546daeab9d76dad20b> [↑](#footnote-ref-1)