CA – Assigment 1: Data Acquisition

- Group: `FakeNews`
- Groupmembers:
  - Adnan Manzoor
  - Sajjad Pervaiz
  - Kevin Taylor
  - Christoph Schäfer

How to reproduce the unified data file:

The unified data is saved in file: `/data/unified_data.json`. To reproduce the file, run the `data-unification.py` in the `code folder` with the following command:

```
python data-unification.py
```

How to run the preliminary statistics:

The methods to calculate the preliminary statistics are located in the file `statistics.py`. If you want to run the file, switch to the `code folder` and execute the the `statistics.py` file with the following command:

```
python statistics.py
```

Otherwise we also added a jupyter notebook file `preliminary-statistics.py`. The jupyter notebook can be opened by the following command

```
jupyter notebook statistics.ipynb
```

Explanation of the method to compute the most specific words of each of the argument units:

To calculate the `most specific words` for the three different argument units, we used the `IF-IDF` score. IF-IDF stands for `Term Frequency – Inverse Document Frequency` and is often used in the information retrieval and text mining. The IF-IDF score/weight 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 (TFIDF.com).
It also can be successfully used for stop-words filtering and other words that appears very often. In our case if a word appears very often in one argument unit but also in other argument unit(s). Therefore we calculate a score for each word based on its relevance and frequency for each of the argument units (major claims, claims and premises).

TF-IDF calculation is defined by

```
IF-IDF := TF * IDF
```

whereas TF and IDF are defined as:

```
IF(t) := (number of times term t appears in a document) /
                    (Total number of terms in a document)
```

and

```
IDF(t) := log_e(Total number of documents /
                Number of documents with term t in it)
```

So in our application:

```
Total number of documents
= total number of essays
= 322
```

And

```
Number of documents with term t in it
= number of essays with term t in it.
```

So for example if a `word w` appears very often in an `argument unit a`, it leads to a high IF score. But if it also appears very often in an other `argument unit a'`, the IDF score will be very low. Therefore the IF-IDF score will be very low in contrast to other words. Consequently we get only a high IF-IDF score, if the frequency of the `word w` is very high and the `word w` is not used very often in other argument units. Therefore, we have calculated the `most specific words` which are very important for each argument unit and also very specifically used for only that argument unit.