Beginning:

Collecting the data from twitter:

After some research we found a file containing all twitter account names for members of the German Bundestag with their respective party. This file had some formatting errors which we had to clean up before using it.

For downloading the required data from twitter we decided to use the opensource framework selenium in combination with google chrome as web browser. This setup allowed us to download as many tweets as we wanted with all of the metadata visible on Twitter.

Because the twitter page is not loaded all at once but continuously as you scroll down, we had to imitate a real user scrolling down the page. For implementing this, we used selenium to scroll to the end of the current view and wait a certain amount of time for the page to load new content. To find the ideal time to wait we had to assess the trade-off between increasing the runtime and the risk of losing some tweets because they weren’t loaded in that short amount of time. We still noticed that we didn’t get all of the tweets in the first run so we decided to do another run in a different network and combine the results.

Here is an overview of the two runs which shows doing a second run increased the total number of tweets by 5%. Because this value is so low, we assume that we got nearly all the data.

For every politician we created an own CSV-file containing the username, post date, text of the tweet, the number of replies, retweets and likes. From these files we created several numpy arrays for training and testing, in which we deleted all retweets to guarantee that all texts were written by the politician.o

223457

235939