Assessing alt-right Twitter use through LSTMs using TensorFlow

# Introduction

The current president of the United States of America Donald J. Trump and his policies have been a controversial topic in public and social media. The social media platform Twitter offers the possibility for individuals to publicly speak their mind about political and social topics regarding these policies.

Distinguishing between political posts and posts that are concerned with other matters is a crucial ability when dealing with social media.

We trained a long short-term memory network with word to vec data generated from tweets. These tweets consisted of political content as well as content unrelated to current political developments. The goal of the training process was to achieve a classifier that can distinguish between political and non-political tweets. Further, we programmed a generator that uses the previously trained classifier to create political tweets.

The from this resulting twitter bot and its tweets can be found here: <https://twitter.com/TrumpFacts8>

# Twitter crawler

Before gathering and filtering tweets, it is crucial to decide which accounts to crawl. This policy aims at obtaining low false-positive and false-negative rates. We manually searched Twitter for accounts that frequently use hashtags related to political opinions. Because some hashtags are being used ironically, we read the most recent tweets of specific twitter accounts and determined if they are posting ironic tweets. Since this pre-selection is based on the author’s political assessment and sensitivity to irony, the resulting list of accounts supporting Trump may be biased.

To obtain tweets we used the tweepy API[[1]](#footnote-1). This library allows to download the latest 3240 tweets from individual accounts in 200-tweet batches. These batches are appended to a list of all tweets from an individual account. Using the json library, tweepy Twitter objects can be filtered for their tweet content, disregarding all additional information such as geolocation, number of retweets etc. The filtered Twitter object content is then stored in a .txt file.

When the crawler has finished collecting tweets from the list of accounts, all twitter messages are merged into one .txt. This step occurs once for the list of accounts of Trump-supporters as well as for accounts posting unrelated content.

These two files are then passed through the preprocessor to obtain the final lists of filtered tweets.

# Preprocessing

Even though Twitter Objects already have been broken down into their text-content, preprocessing is still a vital step to obtain easy to use data. To lessen semantic noise, we filtered tweets that contained any sort of URL from the total list of tweets. Additionally, we filtered tweets that are shorter than 50 characters to increase semantic content of tweets that are being used for the network to learn. Further, some formatting code (for example &amp for the symbol “&”) left in the string of the Twitter Object is erased.

Finally, two .txt files with filtered and preprocessed tweets are created: One from the list of accounts supporting Trump and another from the list of accounts tweeting unrelated content.

1. <http://www.tweepy.org/> [↑](#footnote-ref-1)