Word embeddings for predicting political affiliation based on Twitter data

Ibrahim Abdelaziz[1], Oliver Berg[1], Angjela Davitkova[1], Md Ashraf Hossain[1],
Venkatesh Iyer[1], Shriram Selvakumar[1], Kumar Shridhar[1], and Saurabh
Varshneya[1]

[1]Technische Universität Kaiserslautern

November 21, 2017

1 Introduction

The modern world of social media knows a plethora of means to communicate ones personal opinion and political alignment. With the platform *Twitter*, figures of political interest are expressing their standpoints in small-sized 144 -character texts (recently updated to 280 characters), which contain a comprised message specific to the general public. This yields great potential for automated analysis of party affiliations to classify political persons of interest within the overall political spectrum [?].

Hence, we propose a structured approach of building a deep learning based classification model that utilizes state-of-the-art **word embeddings** [?] to perform **qualitative analysis** on constructed **social media data set**. Our approach will be helpful in analyzing possible early intuitions and dedicated insights within German political spectrum. This is to be seen in context of latest **advances in research**.

2 Data Sets and Feature Extraction

Political motives were shown to be consistently predictable with an accuracy better than chance [?]. According to the presented task description of analyzing political affiliation based on Twitter-data, **tweets** made by political figures on the platform Twitter were used for the findings. Additionally, categorized data set taken from *www.wahl.de/politiker* may be leveraged as a prearrangement of the initial raw Twitter-data.

For the purpose of this specific research, the main portion of data would be the Twitter data. In order to include all of the relevant politicians from the parties, their corresponding Twitter accounts will be collected (Twitter, wahl.de), where the data is offered as open source data. For each of the Twitter accounts of the politicians, a corresponding party is kept, for which the politician is working. Having these accounts, the same number of tweets will be crawled for each of the parties. 80 percent of the collected data would be used from training the classifier and 20 percent of it would be used for testing. Additionally, for more testing purposes, the previously mentioned data sources, that includes parliament discussion data and party manifesto data, can be used.

In order to train models on text, the data needs to be converted into numerical values, more specifically vectors, under specific similarity metrics. One way this can be achieved is to create a word embedding for each of the words from the tweets using *Word2Vec*. While constructing word embeddings, appropriate dimensionality reduction can be applied.

1

3 Recent Developments

Existing papers [**?**] then tackle the classification problem using various techniques, such as SVM, SVD or LSM. Mainly these approaches consider solely the political affiliation in America, where the orientation is rather simpler, since there are only two sides and the users are mainly biased towards one side. Other sources tend to focus on comparison of different classifiers when trying to tackle this problem, and thus not proposing a complete well-developed approach [**?**]. Overall, sentiment classification is mostly covered using recurrent- or convolutional neural networks [**?**].

In connection to the given focus of working on Twitter data, [**?**] introduces objections to some of the preexisting approaches. With standard classifiers for inferring political orientation having greatly lower accuracy from the accuracy that they report, it is stated that the classifiers cannot be used for classifying users outside the training data. Thus the contradictory arguments hold true.

4 Proposed Methodology

Using a recurrent neural network architecture (LSTM) seems a good solution for our specific task. Since the twitter dataset is very complex (giving it's small length and more information), a recurrent neural network alone will be unable to learn all the features. So, a convolutional neural network at the start can be used to learn character wise information before the words are fed to a recurrent neural network that learns the context within data.

The proposed solution will leverage neural networks, where we compare different approaches of both convolutional- and recurrent networks architectures.

5 Analysis of Results

For the results analysis, various metrics can be used with primary ones being Pearson Correlation Coefficient, F1 Score and Cosine Similarities. Further more, clustering of data can be done to find the outliers within the embedding space. Visualization techniques like T-SNE and PCA can be used to visualize the results.