Word embeddings for predicting political affiliation based on Twitter data

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

[1]Technische Universität Kaiserslautern

December 4, 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 [Biessmann et al., 2017].

Hence, we propose a structured approach of building a deep learning based classification model that utilizes state-of-the-art **word embeddings** [Pelevina1a et al., 2016] to perform **qualitative analysis** on constructed **social media data set**. Our approach will be helpful in analyzing possible early intuitions and dedicated insights within the German political spectrum.

This is to be seen in context of latest **advances in research**.

2 Related Work

Political motives were shown to be consistently predictable with an accuracy better than chance [Biessmann et al., 2017].

Many existing papers either propose tackling the classification problem of political bias using techniques such as Support Vector Machine (SVM) or Singular Value Decomposition (SVD) [Misra and Basak, 2016], or they focus on comparison of different classifiers to not restrict themselves to a single well-developed approach [Bhanda et al., 2009]. These approaches mainly consider the political affiliation in America, where the political orientation is mostly binary with two parties - republicans and democrats - covering most of the political spectrum. Overall, sentiment classification is mostly covered using recurrent- or convolutional neural networks [Kim, 2014].

In connection to the given focus of working on Twitter data, [Cohen and Ruths, 2013] introduces objections to some of the pre-existing approaches. With standard classifiers for inferring political orientation having greatly lower accuracy compared to what was initially report, it is stated that the classifiers cannot be used for classifying users outside the training data. Thus the contradictory arguments hold true.

3 Proposed Methodology

As our approach aims to analyze text messages of political figures concerning party affiliation, we leverage *word embeddings* to represent words in context. We shall initially restrict ourselves to a pre-trained model as the number of political parties as classes is not very high. A person's political affiliation will be calculated as a combined analysis of all of his Twitter messages.

Subsequently, a neural network architecture then classifies the Twitter profile, consisting of a collection of this person's tweets, concerning party affiliation. It will learn to represent a political figure's party affiliation through their expression in short message texts.

3.1 Data Sets and Feature Extraction

The main portion of data collected will be Twitter data. In order to include all of the relevant politicians from the parties, their Twitter accounts with their corresponding messages will be collected, where the data is offered as open source data. Each politicians own party is kept as his ground truth class label.

There are seven class labels as there are seven political parties to consider, namely "CDU", "CSU", SPD", "FDP", "GRÜNE", "LINKE" and "AFD", ordered by age of introduction into German parliament, old to new. As such, a pre-trained word embeddings model should suffice to represent textual influence. Alternatively, a custom model may be trained on Wikipedia data dump in German language, possibly in combination with more politically motivated data sources such as German parliament discussion data or party manifesto data.

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* [Mikolov et al., 2013]. While constructing word embeddings, appropriate dimensionality reduction can be applied.

For each party, the same number of accounts and tweets per account will be used to later train the classification model.

Train-test-splits with ration 80-20 will be employed, where the collection of a single person's tweets will be counted as a single data record. It may not be humanly feasible to classify a person's party affiliation based on a single Twitter message, and as such this should also not be the classifier's primary objective.

3.2 Classification

We propose a solution that will leverage neural networks, where we expect to compare both convolutional- and recurrent network architectures for the classification task.

We first aim to learn information of single words through a convolutional neural network (CNN) architecture. This has been proven to yield respectable results [Kim, 2014].

Intuitively though, using a Long Short Term Memory (LSTM) recurrent neural network (RNN) architecture also seems like a good solution for the task at hand. It may smoothly correlate given words with their predecessors to not only take fixed content length into account like feed-forward networks do [Sundermeyer et al., 2012]. Since the twitter dataset is very complex - given its small length and condense information characteristics - a default recurrent neural network alone may not be able to learn the most important features.

4 Analysis of Results

- Original: For the results analysis, various metrics can be used with primary ones being Pearson Correlation Coefficient [Hauke and Kossowski, 2011], 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 [van der Maaten and Hinton, 2008] and PCA [Ric, 2009] can be used to visualize the results.

- Incorporate possible comparison with political compass (https://www.politicalcompass.org/germany2017)

5 Appendix - Work packages and distribution

We plan to distribute the workload into the work packages, for each work package we assign a group of people, and an initial estimated deadline.

Building dataset
Beside the Tweets of German politicians that we will get using the Twitter API, we also plan to collect data from other sources such as parliament discussion data and party manifesto data.

Training word vector model
After building our dataset we will convert the data text into numerically creating word embedding of the words in the text using Word2Vec or derived approaches.

Developing classification model
As previously mentioned in the Proposed Methodology we will implement a neural network as our classifier, and train it with our data set.

Training / Testing
After having quantified model accuracy with a dedicated validation split, this separate training- and testing-stage ensures that the obtained results match initial expectations or reject estimates in an understandable fashion. We thereby ensure that the obtained results obey logic and real-world measures.

Quantitative Analysis
To conclude the findings from real-word social media data analysis, we infer statistical and sociological meaning to the modeled results and put them in context to the initially motivated research question of political affiliation and political classification.

References

[Ric, 2009] (2009). Principal component analysis.

[Bhanda et al., 2009] Bhanda, M., Robinson, D., and Sathi, C. (2009). Text classifiers for political ideologies.

[Biessmann et al., 2017] Biessmann, F., Lehmann, P., Kirsch, D., and Schelter, S. (2017). Predicting political party affiliation from text.

[Cohen and Ruths, 2013] Cohen, R. and Ruths, D. (2013). Classifying political orientation on twitter: It's not easy!

[Hauke and Kossowski, 2011] Hauke, J. and Kossowski, T. (2011). Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data.

[Kim, 2014] Kim, Y. (2014). Convolutional neural networks for sentence classification.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Misra and Basak, 2016] Misra, A. and Basak, S. (2016). Political bias analysis.

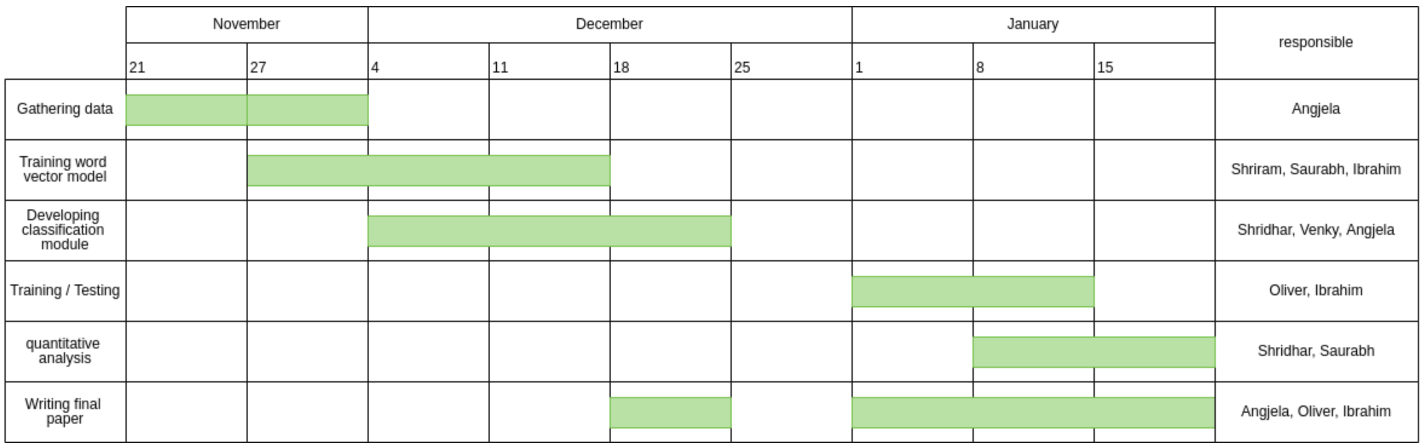

Figure 1: Gantt-Chart displaying workload distribution per team-member

[Pelevina1a et al., 2016] Pelevina1a, M., Arefyev, N., Biemann, C., and Panchenko, A. (2016). Making sense of word embeddings.

[Sundermeyer et al., 2012] Sundermeyer, M., Schlüter, R., and Ney, H. (2012). Lstm neural networks for language modeling. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[van der Maaten and Hinton, 2008] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne.