Predicting political views based on tweets

Florian Siepe
Philipps University of Marburg
Marburg, Germany
Siepe@students.uni-marburg.de

Lennart Hallenberger
Philipps University of Marburg
Marburg, Germany
Hallenb@students.uni-marburg.de

ABSTRACT

In this work, we discuss and evaluated a topic modeling approach towards predicting political views of users based on their tweets. A dataset was used, which contains tweets of U.S. American politicians from the Democratic and Republican Party.

After cleaning and preprocessing the tweets, they are embedded using Latent Dirichlet Allocation (LDA) based on the topics they contain. With these embeddings, classifiers are trained to predict a political party based on a tweet. The results show that no reliably predictions can be made with this embedding technique.

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/Lennart97/nlp_project.

1 INTRODUCTION

Over the past years, social media has become more relevant in politics. Not only are politicians them self's using social media to gain a growing following, it's also where citizens discuss and share their opinion on political topics. From this comes an interest to classify social media posts by their political point of view. One use case can be to gain insights on political topics and their distribution amongst voters. Furthermore, analyzing posts in a large scale can play a role in more accurate projections of future elections.

The goal of this project is to classify users based on their social media postings regarding their political viewpoint to match them with a political party. To test the accuracy of our approach, we use tweets made by members of the Democratic and Republican Party from the United States (2). Next we apply an LDA and train a classifier on these tweets to predict the party based on the tweets content (3). The results from the classifier are then evaluated and discussed (4). Finally, we give a summary and point out the key findings (5).

2 DATASET

The used dataset [1] consists of 84502 tweets from May 2018. These tweets were collected from 433 accounts, whose author are validated members of either the Democratic or Republican Party. The tweets have an equal distribution with 49% being written by Democrats and the remaining 51% by Republicans.

For each account, the last 200 tweets were collected and labeled with their political party.

3 APPROACH

The approach we use to predict the proximity to a political party consists of four main steps.

[1] https://www.kaggle.com/datasets/kapastor/democratvsrepublicantweets

(1) Data cleaning
(2) Preprocessing
(3) Creating an LDA
(4) Classifier training and prediction

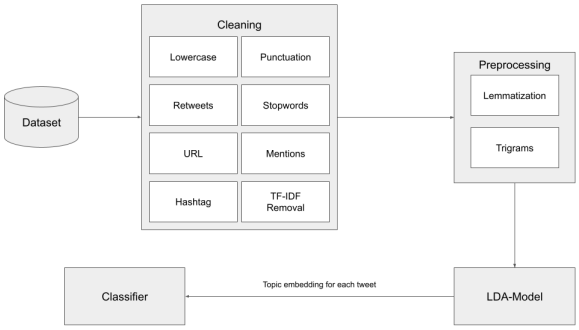


Figure 1: Pipeline of the given approach

Figure 1 illustrates this process. In the following, each of the shown steps will be outlined in greater detail.

3.1 Data cleaning

As tweets are usually noisy in terms of being grammatically correct and often contain human errors like typos, cleaning data from this noise is an essential step.

The cleaning is organized in different modules, each responsible to deal with a certain aspect of the data.

Lowercase. To avoid unnecessary ambiguity, all tokens are converted to lowercase.

Punctuation. Special punctuation characters are removed, as they do not carry semantic meaning.

Stop words. Similarly, stop words like "a" or "the" do not carry semantic meaning either, thus removing them as well.

Retweets. Retweets are a special feature of tweets such that a user repeats a tweet from a different user. However, since it can usually be assumed that retweeting means confirmation with the original tweets, only the respective marker is removed.

URL. Users often provide links to external resources in their tweets. Since extracting topics from these resources is out of scope of this project, URLs are removed from tweets.

Mentions. Twitter users can mention other users. For further improvements this information might be included to strengthen or weaken an existing classification based on the user's political view.

Hashtag. Hashtags express the tweets relevancy for a specific topic. Since users of both party's tweet about the same topics with opposing opinions hashtags appear as noise, thus being removed.

TF-IDF. Some tokens (or words) appear only a few times, while others occur quite often. Either way, this commonly means that these tokens are too specific and might lead to noise in the data, or they do not carry much meaning. Therefore, they might be removed.

3.2 Preprocessing

Having cleaned some aspects of the data, preprocessing is the next step. For this we utilize lemmatization and the creation of trigrams.

Lemmatization. To further reduce ambiguity and complexity of the data all words are being lemmatized and reduced to their base form. While being computationally more expensive, this usually gives better results than stemming, which uses heuristics.

Trigrams. When analyzing word sequences and their meaning the order of words is an important aspect to be considered. To take this into account, we create trigrams to feed them into our model.

3.3 Creating an LDA

Coming back to the main task, to distinguish political views, the essential part is finding features to differentiate between democrat and republican tweets. The difference must be representable in an embedding to train a classifier later for doing the actual classification. The key question is, how to find a unique word or word sequences which are specific for Democrats or Republicans.

In the following, we are outlining an approach using topic modeling. We use the commonly known LDA, a generative probabilistic model [1].

In topic modeling, words (here trigrams) are collected in documents (tweets), where each word's presence is an indicator for a specific topic in the document. On the other hand, each topic also has various words belonging to it. Therefore LDA tries to find topics a document belongs to based on the word in the document.

While LDA is an unsupervised learning method, the number of desired topics k needs to be specified upfront. Thus, another challenge is to find a k where the distance between topics which are assigned to Democrats and Republicans is maximized. Section 4 further explains how an optimal k is found.

3.4 Training a classifier

After creating a topic model on our data, the next step is training a classifier.

Clearly, an embedding has to be created upfront for our tweet dataset. For this, we utilize the LDA we created before. For each tweet the LDA is applied, which outputs a likelihood distribution over the topics, that indicates the likelihood for a tweet to contain specific topics. The output of the LDA is a vector $\vec{v}$ with $|\vec{v}| = k$.

With this embedding, a classifier is trained against the respective labels "Democrat" and "Republican". To see how different classifier perform, we evaluate *Gaussian Naive Bayes (G-NB)*, *Linear Support Vector Machine (L-SVM)* and *Multi-Layer Perceptron (MLP)*.

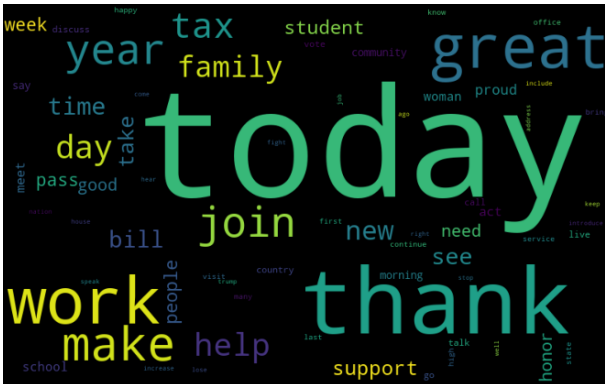


Figure 2: Words removed by TF-IDF

4 EVALUATION

Figure 2 shows the words removed using TF-IDF during the data cleaning. The size of a word represents how often it was removed from tweets overall. Especially the biggest words shown, like "today", "thank" and "work" were often present in tweets by Democratic or Republican Party members. Removing these words resulted in a higher differentiation between topics by the LDA because of less overlapping words.

As outlined in 3.3 a topic count k needs to be found. Therefore we created models with an increasing number of topics and measure the associated logarithmic perplexity and coherence (UMass) of the models.

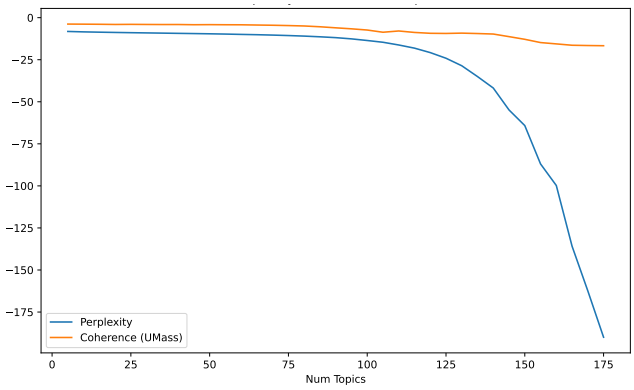


Figure 3: logarithmic perplexity and coherence over k

Figure 3 shows the perplexity and coherence over k. Measurements of both values were collected starting with $k = 5$ increasing the number of topics by 5 till $k = 175$. We can observe a decline in perplexity and coherence with an increasing number of topics. While we see a slow but steady decline from the start, the perplexity drops significantly with $k > 120$.

Similarly, the performance of MLP, G-NB and L-SVM have been observed over the range of topics from $k = 5$ to $k = 175$ in steps of 5. With the results we can determine the optimal number of topics k and the best performing classifier.

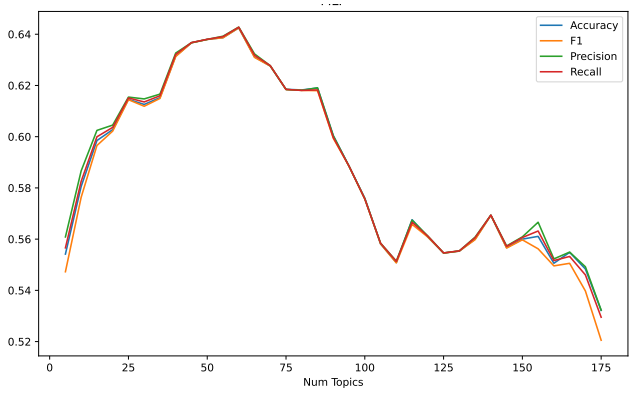


Figure 4: Performance of MLP

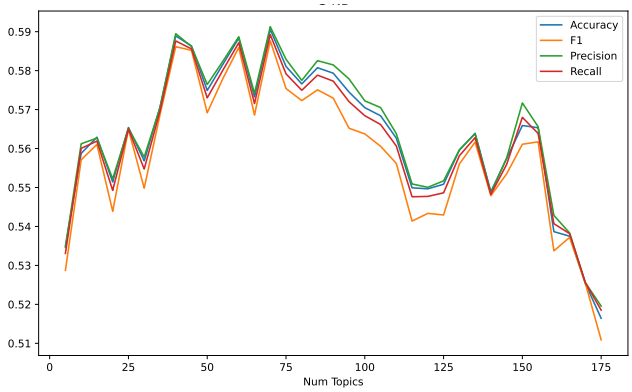


Figure 5: Performance of G-NB

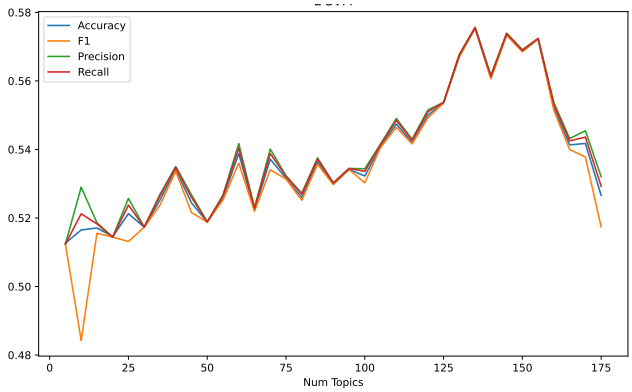


Figure 6: Performance of L-SVM

In the range from $k = 5$ to $k = 100$, we can see a clear advantage of the MLP over the other classifiers. The MLP classifier also has the overall highest accuracy at $k = 60$. A similar behavior but with an overall lower score can be seen for the G-NB. The L-SVM is showing the worst performance rising from the start with a peak at $k = 130 - 150$ where a rapid drop of perplexity was observed previously.

During the comparison, all classifiers were trained with their default parameter settings. After determining MLP as the best performing classifier, hyperparameter optimization was performed using grid search.

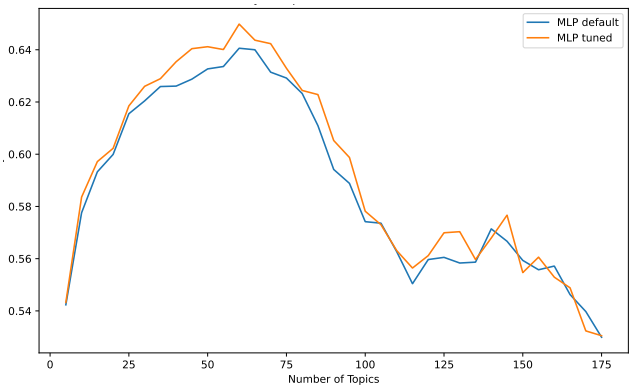


Figure 7: MLP accuracy default and tuned parameters

Figure 7 shows a comparison between a MLP classifier with default parameters and the tuned parameters from the grid search. With the tuned parameters an overall improvement can be observed. As before with default parameters the highest accuracy can be seen at $k = 60$.

A further factor with a big impact on the performance of all classifiers is the amount of tweets used.

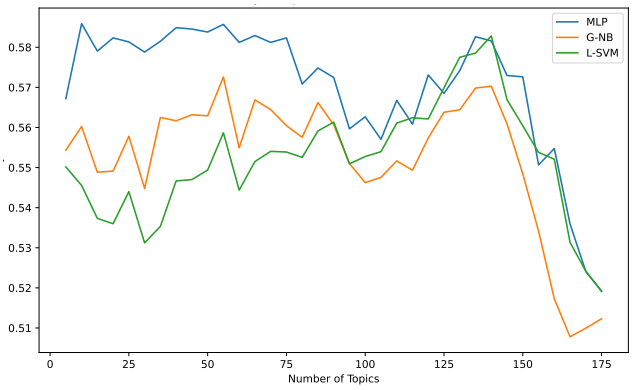
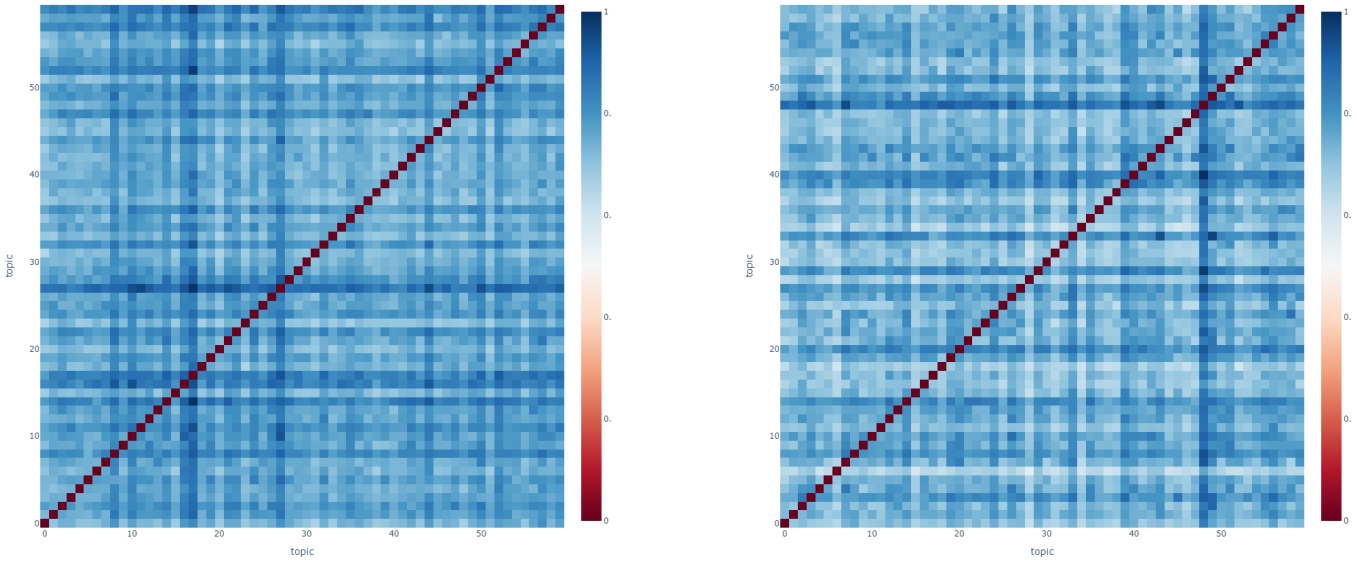


Figure 8: Accuracy with 80,000 tweets

For the previous performance measurements 40,000 tweets were used, with an even distribution between members of the Democratic or Republican Party. In figure 8 an overall lower accuracy for all classifiers is shown using 80,000 tweets.



(a) 40,000 tweets **(b) 80,000 tweets**

Figure 9: Pairwise dissimilarity of 60 topics created with differing number of tweets

The drop in accuracy with increased tweets can be explained with a lower dissimilarity between topics. Figure 9 is showing a comparison between two dissimilarity matrices from LDAs with 60 topics using 40,000 and 80,000 tweets. The overall lighter blue of the matrix in 9b indicates a lower distinction between topics. Therefore vector embeddings becoming more homogeneous, resulting in lower performance of the used classifiers.

In conclusion, the best performance was found with 40,000 tweets using MLP classifier with tuned parameters achieving an accuracy of 0.651.

The shown approach presumes the same LDA is used for embedding the training and test data. When splitting the data and creating separate LDAs for test and training, the accuracy drops to 0.50 using the MLP classifier.

5 SUMMARY

To summarise the findings of this approach it is clear, that using LDA as embedding for classifiers does not produce sufficient results. Even in this best-case scenario having only two parties and tweets from politicians rather than citizens it was not possible to obtain reliable predictions of the author's political view.

Therefore, it seems that unsupervised, generative LDAs are unable to capture semantic meaning towards certain, for the political position relevant topics, by using statistical methods.

Further work could investigate, whether supervised methods, especially Deep Learning, bare better solutions to this task. For example one may fine tune existing and overall well performing transformer models such as BERT for this classification task.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (mar 2003), 993–1022.