Predicting political party affiliation from text

Felix Bießmann[*] **Daniel Kirsch**[†]

Abstract

In this study we investigate, to what extent political party affiliation can be predicted from textual content. We used the text of speeches and discussions in the German parliament to train a classifier that predicts the political party affiliation based on standard text features. The classifier is evaluated on parliament speeches and party manifestos. Results indicate that automatic classification of political affiliation is possible with well above chance accuracy. Sentiment analysis shows political power, measured in terms of seats in the parliament, correlates with the sentiment of a political speech: speeches are more positive when a party is in power and more negative when the text is from a member of the opposition. These results can be considered a first step towards automatised extraction of political views, a necessary requirement for unbiased political education in the presence of massive amounts of media content. Our model can readily be used to analyse texts for which the party affiliation is not clear, such as news articles or texts from lobby groups.

1 Introduction

Estimating the political view expressed by some content can be difficult if the speaker and his political affiliation is not known. Extracting political views from texts however can be crucial for a variety of applications. Two examples have become particularly important in recent years. For one social networks are increasingly being used for recruitment and propagation of inappropriate content by members of extreme political groups. Detection of these posts and individuals requires knowing their political affiliation. A second example is political education. As democracy becomes a more popular form of government, it becomes more important for the parties in power to steer the public opinion. The common way groups in power do this is to spread their views via media content. A sensible strategy to immunise people against biased content is to consume a heterogeneous mix of content from a multitude of political views. This approach however requires to be able to attribute content to a political view. Even human experts are often challenged when confronted with this task. But in the presence of an ever growing flow of media content, it is difficult to imagine that the content published on a daily basis on the internet can be attributed to political views by human labelling. This in turn highlights the need for automatisation of this process, in order to tackle not only the two above mentioned problem settings.

We here propose a method for automatically associating content to political parties. Our approach trains a model on standard text features from speeches held in parliament discussions [3] and the corresponding party as label. Next to this four class classification we also looked at a slightly simpler binary classification problem, in which we train a classifier to discriminate texts from members of the government and members of the opposition. As generalization performance of such an automatized system for political party prediction is difficult to measure, we also compare how well the learned models perform on the manifestos which the parties released before this governmental period [7, 4, 11, 5].

[*] felix.biessmann@gmail.com
[†] mail@danielkirs.ch

In order to interpret the features used by the model we also look at the correlations between the text features used by the model and the party label. We conjecture that this approach can yield some insights as to which discriminative features can be used by laymen to tell the political affiliation from texts that do not have clear party affiliations.

Finally we were interested in whether political power, measured in terms of seats in the parliament that a given party has, has an influence on the content, in particular the sentiment, of a speech. To this end we performed analyses that quantify whether the text of what a member of a given party said contains rather positive or negative sentiment. This analysis can be seen as a quantitive analysis of the common sense intuition that helps us to critically assess textual media content.

2 Methods

In the following we list the steps we performed in order to learn a model that automatically attributes texts to parties. While not all of the experiments we tried are listed below, we made the code available on GitHub [2]. The code should allow to download all data sources and run all experiments performed in this study. For more lightweight testing of the presented results we also provide a demo web site on which the model can be tested with plain text files or links to webpages containing the to-be-categorized text [1].

2.1 Data Collection

We extracted the texts from the official webpage of the German parliament [3] in the period from August 2013 to August 2015. The texts were split using regular expressions and party labels were extracted from the texts. The number of speeches for each political party is listed in Table 1.

Next to the parliament speeches we also extracted the text from the programs of each party for this legislative period [7, 4, 11, 5] and applied exactly the same preprocessing to these texts that we applied to the parliament speeches. Texts were split according to chapters in the respective texts, which resulted in an asymmetric number of data points per party, see also Table 1.

2.2 Data Cleaning

Prior to vectorization of the text data, we cleaned the texts with a variety of heuristics and stopwords. We did so to avoid biasing our classifier towards using words that are helpful for classifying parliament speeches but do not carry information about the actual target, political orientation[1]. We first removed stopwords from the text, using a stopword list [10] and additional stopwords common to parliament speeches. Another option we considered for cleaning the text was to include all names of members of the parliament in the stopword list. Of course also the names of the parties themselves were removed from the text. Moreover we removed numbers and non-word symbols from the text. Finally we also removed sections from the texts that were not part of a speech and parts in the speeches that were short interruptions (applaus or critical remarks from other members of the parliament). As we did not have clear ideas what of these procedures was actually helpful for improving generalization performance, we optimized the cleaning procedure by including all options of the cleaning pipeline into the hyperparameter optimization, see subsection 2.5.

2.3 Bag-of-Words Vectorization

Strings of teach speech are extracted using a bag-of-word vectorizer as implemented in scikit-learn [8]. The general idea of bag-of-words vectors is to simply count occurences of words (or word sequences, also called *n-grams*) for each data point (usually documents, here parliament speeches). So the text of each speech is transformed into a vector $\mathbf{x} \in \mathbb{R}^d$, were d is the size of our dictionary; the wth entry of $\mathbf{x}$ contains the (normalized) count of the wth word (or sequence of words) in

[1] A classical example from computer vision of such kind of overfitting with machine learning methods was the winning approach of a 'tank-detection' challenge. On all images in the training set, the tank was depicted on a green field in front of blue sky, the negative examples did not feature this surrounding, so the classifier learned to detect green fields and blue sky and could perfectly detect tanks in this biased data set.

our dictionary. We tried several options for vectorizing the speeches, including term-frequency-inverse-document-frequency normalisation, stemming and using n-gram patterns up to size $n = 5$ (including hashing to a dictionary size of $d \approx 10^6$). All of these hyperparameters were subjected to hyperparameter optimization as explained in subsection 2.5.

2.4 Classification

We used a multinomial logistic regression model in order to classify bag-of-words feature vectors $\mathbf{x} \in \mathbb{R}^d$, where d is the size of the bag-of-words dictionary. Let $y \in \{1, 2, \ldots, K\}$ be the true party label, where K is the total number of parties in the parliament and $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{d \times K}$ be the concatenation of the weight vectors $\mathbf{w}_k$ associated with the kth party then

$$p(y = k|\mathbf{x}, \mathbf{W}) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \qquad \text{with } z_k = \mathbf{w}_k^\top \mathbf{x} \tag{1}$$

We estimated $\mathbf{W}$ using quasi-newton gradient descent as packaged in the scikit-learn toolbox. The optimization function was obtained by adding a penalization term to the negative log-likelihood of the multinomial logistic regression objective and the optimization hence found the $\mathbf{W}$ that minimized

$$L(\mathbf{W}, \mathbf{x}, \gamma) = -\log \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} + \gamma \|\mathbf{W}\|_F \tag{2}$$

Where $\| \ \|_F$ denotes the Frobenius Norm and γ is a regularization parameter controlling the complexity of the model. We optimized the regularization parameter on a log-scaled grid from $10^{-4,\ldots,4}$ and the regularization constant was adopted to reflect asymmetric class frequency distributions. The performance of the model was optimized using the classification accuracy, but we also report all other standard measures, precision ($TP/(FP + TP)$, recall ($TP/(TP + FN)$) and f1-score ($2 \times (Prec. \times Rec)/(Prec + Rec.)$).

Next to the four class problem of categorizing all four parties of the german parliament we also considered the binary classification task of only distinguishing texts between wether they were read by members of the government or members of the opposition. Results of this simpler two-class problem were obtained by exactly the same methods as in the four class case.

2.5 Optimisation of Model Parameters

The model pipeline contained a number of *hyperparameters* that we did not want to tune by hand. These parameters are called hyperparameters becase we cannot optimize these like we optimize the parameters of our classifier, by writing down an objective function and optimizing this linear function by doing gradient descent. Thus we resort to the standard 'trial-and-error' approach in machine learning and perform cross-validation on the hyperparameters. In order to not overfit to our data and obtain good generalization performance to other texts we used two approaches. The first one is the usual procedure of 'nested-crossvalidation', which performs two stages of cross-validation, an *outer* cross-validation loop to evaluate the model correctly on held-out data, and an *inner* cross-validation loop within each training data fold of the outer cross-validation. In practice this means that in each step of the outer loop we set aside a part of our data set and then try out all parameters using the inner cross-validation on the not-held-out data, and afterwards make a prediction on the held-out data of the outer cross-validation. For each set of preprocessing (hyper-) parameters and regularisation parameter of the classifier, we performed the entire data extraction, vectorization and classifier training workflow. Next to this rather standard approach, we also tested the classifiers on the text of the party programs [7, 4, 11, 5], which were never included in any of the training procedures.

2.6 Sentiment analysis

We used a publicly available key word list to extract sentiments from the text subjected to classification [9]. The sentiment index used for attributing positive or negative sentiment to a text was computed as the Pearson correlation coefficient between bag-of-word vectors $\mathbf{x} \in \mathbb{R}^d$ and sentiment vector $\mathbf{s} \in \mathbb{R}^d$ which was constructed by simply taking the values from the sentiment dictionary.

	Held-out parliament speeches				Party Manifestos			
	precision	recall	f1-score	N	precision	recall	f1-score	N
Linke	0.61	0.61	0.61	2315	0.81	0.76	0.79	104
Grüne	0.65	0.64	0.64	3050	0.57	0.21	0.31	110
SPD	0.55	0.55	0.55	2608	0.10	0.50	0.16	16
CDU/CSU	0.68	0.70	0.69	3837	0.06	0.08	0.07	38
avg / total	0.63	0.63	0.63	11810	0.57	0.42	0.45	268

Table 1: Classification performance on the four party prediction problem; while classification on held-out data from parliament speeches was on average higher than on party programs, also on the completely out-of-training-sample data the model obtained reasonable accuracies in some cases.

	Held-out parliament speeches				Party Manifestos			
	precision	recall	f1-score	N	precision	recall	f1-score	N
Opposition	0.84	0.82	0.83	5365	0.99	0.70	0.82	214
Government	0.85	0.87	0.86	6443	0.45	0.98	0.62	54
avg / total	0.85	0.85	0.85	11808	0.88	0.75	0.78	268

Table 2: Classification performance on the binary prediction problem, categorizing speeches to government (SPD/CDU/CSU) and opposition (Linke, Grüne).

2.7 Analysing bag-of-words features

In order to get better picture of which features contribute to the classification we computed correlation coefficients between each word and each party. Often linear models are interpreted by just looking at the weights of the model. Note that due to the fact that the features are obtained from real data and thus correlated we cannot inspect the model by interpreting the coefficients of $\mathbf{W}$; for an in-depth discussion of this problem see e.g. [6]. The words corresponding to the top positive and negative correlations are shown in section section 5.

3 Results

After optimizing the model by means of an exhaustiv grid search over all preprocessing parameters we finally ended up with an optimal parameter setting for both conditions, binary classification (government vs opposition) and the four class problem, predicting all four parties separately. The optimal parameter settings were chosen to maximize the generalization performance of the model on both the parliament speeches and the party manifestos. We chose the parameter setting that gave accuracy by averaging the accuracies for both test data sets.

3.1 Classification performance

Table 1 shows the performance of the model that was trained on predicting all four parties. The best parameter setting included stemming, stopword removal as well as removal of interruptions such as applause, followed by unigram extraction and tfidf normalisation, yielding an average accuracy of 0.63 on held-out parliament speeches. Inspecting the results shown in Table 1 we note that in both test cases, prediction on parliament speeches as well as predictions on party manifestos, the model performs clearly well above chance performance, which would correspond to an accuracy of 0.25, assuming equal sizes of classes.

Table 2 shows the results for the binary classification setting. Here the optimal parameters were found to be a very simple preprocessing, consisting of stopword removal and unigram extraction, reaching an average accuracy of 0.85 (held-out parliament speeches) and 0.78 (held-out party manifestos).

While the results on parliament speeches, the data on which the classifiers were trained, do not exhibit any interesting structure across classes (party or government affiliation), the classification accuracy on the party manifestos does show a systematic pattern: While party programs of the

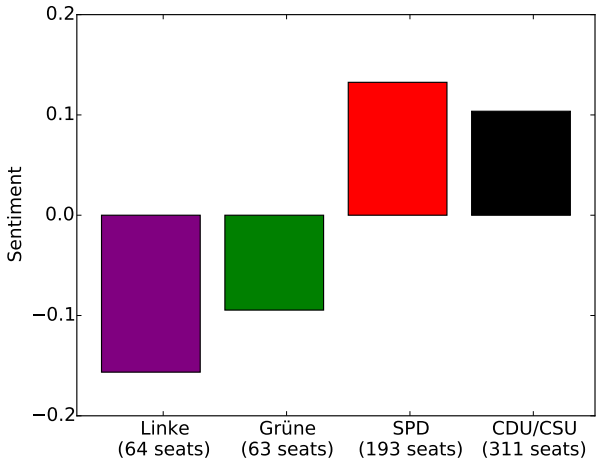

Figure 1: Speech sentiments computed for speeches of each party (parties are ordered according to political orientation from left to right). The more political power a party has (power is measured in seats in the parliament, respective seats are given in the figure), the more positive the speech content is, the fewer seats a party has the more words associated with negative emotions are used in a speech.

opposition, especially of the left wing, can be classified relatively reliably with precision and recall of 0.81 and 0.76, respectively, the party manifestos of the government, especially the strongest party of the christian conservatives, cannot reliably predicted from the model trained on the parliament speeches. This can have a number of reasons, one being that there is a covariate shift between the text of the party manifestos, written before the election, and the texts of the parliament speeches.

4 Speech sentiment correlates with political power

We analysed the sentiment of each speech of a political party and plot the results in Figure 1. While the methodology used for estimating the sentiment (see subsection 2.6), it can yield some insight into the emotional content of a speech. High values indicate more pronounced usage of positive words, whereas negative values indicate more pronounced usage of words associated with negative emotional content. We can quantify political power as the number of seats a party has in the parliament. Our analyses show a strong trend supporting the hypothesis that the more political power a party has, the more positive sentiment their speech has, and vice versa, the less seats a party has, the more negative content their speeches contain. This pattern is evident from the barplot in Figure 1 and supported by a high correlation of 0.86 between number of parliament seats and sentiment index.

5 Correlations between words and parties

Figure 2 shows the correlations of individual words and each party. We plotted the top 20 words for each sign, i.e. the 20 words with the strongest positive and negative correlations for each party, respectively. While the correlations do indicate that there are – despite extensive and careful cleaning procedures – still words with strong correlations in the top words, we conjecture that at least some of these words do have some discriminatory power, even if they do not refer to political views. Below we list some examples, but the most prominent is probably: the opposition often addresses the government in their speeches, the govermental parties do that less often. This is reflected in the frequency of the usage of words that refer to members of the government or ministers. But among the often used and avoided words we also find words that refer to actual political content. Thus we hypothesize that to some extent these correlation coefficients can give hints as to what the bag-of-words histogram representation into which we transformed the speeches are related to the actual content of the speeches. In the following we give some examples of words that appear to be preferentially used or avoided by each respective party. Even though interpretations of these

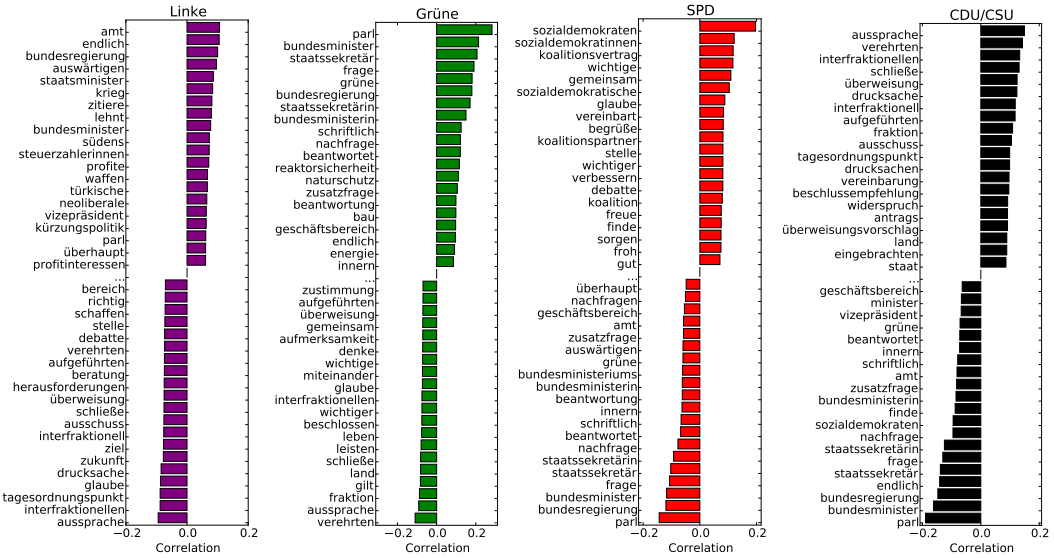

Figure 2: Correlations between words and party for parliament speeches.

quantitative results are somewhat invalid, as they neglect the context in which these words were mentioned, we find some interesting patterns.

Linke (left opposition) The left wing opposition rarely mentions the daily agenda or interfraktional work; while the left opposition very often addresses the government and the ministers, as indicated by the words *Bundesregierung* (eng.: federal government), *Staatsminister* (engl.: federal minister), these addresses avoids the respectful introduction *verehrten*. Positively correlated words include *endlich* (engl.: finally), puts the focus on *Steuerzahler* (engl.: taxpayers), and *neoliberale* (engl.: neoliberal) attitudes that would promote *Kürzungspolitik* (engl: cutting of investments in healthcare and social wellfare) while not being very sceptical towards ethically dubious economic practices, as alluded to by the words *Waffen* (engl.: Weapons) or *Profite* (engl,: profit) and *Profitinteressen* (engl: profit interests). Among the most often mentioned word in speeches of the left opposition is the word *Krieg* (engl.: war) – the left opposition is often the first party to use this word and to advocate diplomatical alternatives, when other parties do not use this word at all and periphrase the same events with other words.

Grüne (green/liberal opposition) Just like the left opposition also the more liberally minded green opposition very often addresses the federal government. The next-most likely words are however in contrast to the left opposition not words that refer to actual political content, but rather to words like *Nachfrage* (engl.: question) or *Zusatzfrage* (engl.: additional question), indicating that these formal steps to be taken by a party of the Bundestag to question certain political decisions of the government were indeed undertaken by the green party and that the party is talking about these inquirements. As to be expected from the political program of the green party, among the most often used words in their speeches include words related to ecological concerns, such as *Reaktorsicherheit* (engl.: security of nuclear plants), *Naturschutz* (engl.: environmental activities) and *Energie* (engl.: energy).

SPD (social democrats) While we did filter out names of representatives in the parliament and of parties, some words that are part of a party name, like *Sozialdemokraten* (engl.: social democrats) are more general than the reference to that party only. As this word relates to political content, too, and in particular to the one represented by this party, it is often used in their speeches. the next most often used, interestingly, is *Koalitionsvertag* (engl.: contract of governmental coalition). The fact that the social democrats use this word frequently could be interpreted as an apologetic argumentation strategy, in the sense of 'we would love to enforce more socialdemocratic decisions, but we have to follow the contract'. In general the most often used words indicate a certain tendency

for euphemistic formulations in the SPD, amongst the most often used words are *begrüße* (engl.: to welcome), *freue* (engl.: being glad), *froh* (engl.: happy), *gut* (engl.: good). Rarely used words are often related to the word *frage* (engl.: question), which can be interpreted as a more affirmative stance towards the work of the parliament.

CDU/CSU (christian conservatives) While the oppositional parties have amongst their most often used words some that do refer to actual political content, the largest party in the government almost has almost exclusively words that relate to formal procedures in their top words list. An often used word is *Beschlussempfehlung* (engl.: recommendation for a decision), which refers to the work of commissions that give recommendations for political decisions, another often used word is *Tagesordnungspunkt* (engl.: item on an agenda); also the focus is often put on previous *Vereinbarung[en]* (engl.: agreements) and *interfraktionell* (engl. between political parties) work. The less often used words include addresses to the governmental staff or words relating to questions.

6 Conclusion

In this study we proposed a system that automatically associates texts with political parties. We performed extensive model selection on all parameters of the feature extraction as well as the model training and found that using rather simple methods we can obtain classification performances well above chance performance for both party prediction as well as prediction of membership of the government or the opposition. We emphasize that this is merely a first step necessary to automatise extraction of political views from textual content. Such an automation is a requirement for a more unbiased political education when confronted with massive amounts of media content. Future research can build on these findings in order to extract political affiliation from other text sources, such as news articles, or texts published by lobby groups that strive to influence political parties.

References

[1] Felix Biessmann and Daniel Kirsch. Political party affiliation prediction. `http://www.linksrechts.kirelabs.org`, 2015.

[2] Felix Biessmann and Daniel Kirsch. Political party affiliation prediction github repository. `http://github.com/kirel/political-affiliation-prediction`, 2015.

[3] Deutscher Bundestag. Plenarprotokolle des deutschen bundestags. `http://www.bundestag.de/plenarprotokolle`, 2013-2015.

[4] Bündnis 90/Grüne. Parteiprogramm. `http://www.gruene.de/fileadmin/user_upload/Dokumente/Wahlprogramm/Wahlprogramm-barrierefrei.pdf`, 2013.

[5] CDU/CSU. Parteiprogramm. `http://www.cdu.de/sites/default/files/media/dokumente/regierungsprogramm-2013-2017-langfassung-20130911.pdf`, 2013.

[6] Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Biessmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87:96–110, 2014.

[7] DIE LINKE. Parteiprogramm. `https://www.die-linke.de/fileadmin/download/wahlen2013/bundestagswahlprogramm/bundestagswahlprogramm2013_langfassung.pdf`, 2013.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] R. Remus, U. Quasthoff, and G. Heyer. Sentiws – a publicly available german-language resource for sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation (LREC'10)*, 2010.

[10] Solaris. Stopword list. `http://solariz.de/649/deutsche-stopwords.htm`.

[11] SPD. Parteiprogramm. `https://www.spd.de/linkableblob/96686/data/20130415_regierungsprogramm_2013_2017.pdf`, 2013.