

Optimizing text representations to capture (dis)similarity between political parties

Tanise Ceron[△] Nico Blokker[□] Sebastian Padó[△]

[△] Institute for Natural Language Processing, University of Stuttgart, Germany

[□] Research Center on Inequality and Social Policy, University of Bremen, Germany
{tanise.ceron,pado}@ims.uni-stuttgart.de, blokker@uni-bremen.de

Abstract

Even though fine-tuned neural language models have been pivotal in enabling “deep” automatic text analysis, optimizing text representations for specific applications remains a crucial bottleneck. In this study, we look at this problem in the context of a task from computational social science, namely modeling pairwise similarities between political parties. Our research question is what level of structural information is necessary to create robust text representation, contrasting a strongly informed approach (which uses both claim span and claim category annotations) with approaches that forgo one or both types of annotation with document structure-based heuristics. Evaluating our models on the manifestos of German parties for the 2021 federal election. We find that heuristics that maximize within-party over between-party similarity along with a normalization step lead to reliable party similarity prediction, without the need for manual annotation.

1 Introduction

A party manifesto, also known as electoral program, is a document in which parties express their views, intentions and motives for the next coming years. Since this genre of text is written not just to inform, but to persuade potential voters that the parties compete for (Budge et al., 2001), it provides a strong basis to understand the position taken by parties according to various policies because of its direct access to the parties’ opinions. Political scientists study the contents of party manifestos, for instance, to investigate parties’ similarity with respect to the several policies (Budge, 2003), to predict party coalitions (Druckman et al., 2005), and to evaluate the extent to which the parties that they vote for actually corresponds to their own world view (McGregor, 2013).

To carry out systematic analyses of party relations while taking into account differences in style and level of detail, these analyses are increasingly

grounded in two types of manual annotation about *claims*, statements that contain a position or a view towards an issue, that can be argued or demanded for (Koopmans and Statham, 1999): First, *abstract claim categories* (Burst et al., 2021) are used to group together diverse forms and formulations of demands. Second, annotation often includes the *stance* that parties take towards specific political claims to abstract away from the many ways to express support or rejection in language. In addition, these types of annotation offer a direct way to empirically ground party similarity in claims and link these to concrete textual statements. At the same time, such manual annotation is extremely expensive in terms of time and resources and has to be repeated for every country and every new election.

In this paper, we investigate the extent to which this manual effort can be reduced given appropriate text representations. We build on the advances made in recent years in neural language models for text representations and present a series of fine-tuning designs based on manifesto texts to compute party similarities. Our main hypothesis is that the proximity between groups can be more easily captured when the model receives adequate indication of the differences between groups (and their stances) and this can be done via fine-tuning for instance. This can be achieved by using signal that is freely available in the manifestos’ *document structure*, such as groupings by party or topic. Information of this type can serve as an alternative feedback for fine-tuning in order to create robust text representations for analysing party proximity.

We ask three specific questions: (1) How to create robust representations for identifying the similarity between groups such as in the case of party relations? (2) What level of document structure is necessary for this purpose? (3) Can computational methods capture the relation between parties in unstructured text? We empirically investigate these questions on electoral programs from the Ger-

man 2021 elections, comparing party similarities against a ground truth built from structured data. We find that our hypothesis is borne out: We can achieve competitive results in modelling the party proximity with textual data provided that the text representations are optimized to capture the differences across parties and normalized to fall in a certain distribution that is appropriate for computing text similarity. More surprisingly, we find that completely unstructured data reach higher correlations than more informed settings that consider exclusively claims and/or their policy domain. We make our code and data available for replicability.¹

Paper structure. The paper is structured as follows. Section 2 provides an overview of related work. Section 3 describes the data we work with and our ground truth. Section 4 presents our modeling approach. Sections 5 and 6 discuss the experimental setup and our results. Section 7 concludes.

2 Related Work

2.1 Party Characterization

The characterization of parties is an important topic in political science, and has previously been attempted with NLP models. Most studies, however, have focused on methods to place parties along the left to right ideological dimension. For instance, an early example is [Laver et al. \(2003\)](#) who investigate the scaling of political texts associated with parties (such as manifestos or legislative speeches) with a bag of words approach in a supervised fashion, with position scores provided by human domain experts. Others, instead, have implemented unsupervised methods for party positioning in order to avoid picking up on biases of the annotated data and to scale up to large amounts of texts from different political contexts while still implementing word frequency methods ([Slapin and Proksch, 2008](#)). More recent studies have sought to overcome the drawbacks of word frequency models such as topic reliance and lack of similarity between synonymous pairs of words, e.g. [Glavaš et al. \(2017\)](#) and [Nanni et al. \(2022\)](#) implement a combination of distributional semantics methods and a graph-based score propagation algorithm for capturing the party positions in the left-right dimension.

Our study differs from previous ones in two main aspects. First, our aim is not to place parties a

¹https://github.com/tceron/capture_similarity_between_political_parties.git

left-to-right political dimension but to assess party similarity in a latent multidimensional space of policy positions and ideologies. Second, our focus is not on the use of specific vocabulary, but on representations of whole sentences. In other words, our proposed models work well if they manage to learn how political viewpoints are expressed at the sentence level in party manifestos.

2.2 Optimizing Text Representations for Similarity

Fine Tuning. Recent years have seen rapid advances in the area of neural language models, including models such as BERT, RoBERTa or GPT-3 ([Devlin et al., 2019](#); [Liu et al., 2020](#); [Brown et al., 2020](#)). The sentence-encoding capabilities of these models make them generally applicable to text classification and similarity tasks ([Cer et al., 2018](#)). Both for classification and for similarity, it was found that pre-trained models already show respectable performance, but fine-tuning them on task-related data is crucial to optimize the models' predictions – essentially telling the model which aspects of the input matter for the task at hand.

On the similarity side, a well-known language model is Sentence-BERT [Reimers and Gurevych \(2019\)](#), a siamese and triplet network based on BERT ([Devlin et al., 2019](#)) or RoBERTa ([Liu et al., 2020](#)) which aims at better encoding the similarities between sequences of text. Sentence-BERT (SBERT) comes with its own fine-tuning schema which is informed by ranked pairs or triplets and tunes the text representations to respect the preferences expressed by the fine-tuning data. Of course, this raises the question of how to obtain such fine-tuning data: The study experiments both with manually annotated datasets (for entailment and paraphrasing tasks) and with the use of heuristic document structure information, assuming that sentences from the same Wikipedia section are semantically closer and sentences from different sections are further away. Parallel results are also found by [Gao et al. \(2021\)](#) in their SimCSE model, which reach even better results when fine-tuning with contrastive learning: They also compare a setting based on manually annotated data from an inference dataset with a heuristic setting based on combining a pair of sentences with its drop-out version as positive examples and different pairs as negative examples.

Both studies find slightly lower performance for

Party	Sentence	Domain
AfD	People’s insecurities and fears, especially in rural regions, must be taken seriously.	Social Groups
CDU	We want to strengthen our Europe together with the citizens for the challenges of the future.	External Relations
Linke	The policies of federal governments that ensure private corporations and investors can make big money off our insurance premiums, co-pays and exploitation of health care workers are endangering our health!	Political System
FDP	In this way, we want to create incentives for a more balanced division of family work between the parents.	Welfare and Quality of Life
Grüne	After the pandemic, we do not want a return to unlimited growth in air traffic, but rather to align it with the goal of climate neutrality.	Economy
SPD	We advocate EU-wide ratification of the Council of Europe’s Istanbul Convention as a binding legal norm against violence against women.	Fabric of Society

Table 1: Examples from the 2021 party manifestos and their annotated domains.

the heuristic versions of their fine-tuning datasets, but still obtain a relevant improvement over the non-fine-tuned versions of their models, pointing to the usefulness of heuristically generated fine-tuning data, for example based on document structure.

Postprocessing to Improve Embeddings A problem of the use of neural language models to create text representations that was recognized recently concerns the distributions of the resulting embeddings: They turn out to be highly anisotropic (Ethayarajh, 2019; Gao et al., 2019), meaning that their semantic space takes a cone rather than a sphere format - in the former two random vectors are highly correlated while in the latter they should be highly uncorrelated. This can cause similarities between tokens or sentences to be very similar even when they should not. To counteract this tendency, Li et al. (2020) impose an isotropic distribution onto the embeddings via a flow-based generative model. Su et al. (2021) propose a lightweight, even slightly more effective approach: The text embeddings undergo a linear so-called whitening transformation, which ensures that the bases of the space are uncorrelated and each have a variance of 1.

3 Data

Before we describe the methods we will use, we describe our textual basis and the ground truth we will aim to approximate.

3.1 The Manifesto Dataset

As stated above, we are interested in deriving party representations from party manifestos. Party mani-

festos generally contain sections roughly separated by policy topics, however, some party manifestos are organized more strictly by topics than others. For this reason, we utilize the manifesto dataset provided by the Manifesto Project (Burst et al., 2021), which provides manifestos from around the world and offers consistent markup of policy domains and categories ².

More specifically, every sentence from the manifestos is annotated with domain names and categories. In this paper, consistent with our goal of reducing annotation effort, we consider only the domain. The domain corresponds to a broad policy field such as ‘political system’ and ‘freedom and democracy’. In most cases, an entire sentence is annotated with a single domain, but some sentences have been split when falling into two distinct domains. Nearly every sentence is annotated with a domain label, except the introduction and end sections which usually contain an appeal to the voter and do not belong to any policy category.

For reasons that will become clear in the next subsection, we focus on German data and use the party manifestos written by the six main German parties (CDU/CSU, SPD, Grüne, Linke, FDP, AfD) for the federal elections in 2013, 2017 and 2021. Table 1 shows some examples of sentences with their respective domain names. Due to space constraints, more information about the description of the dataset is found in appendix A.1.

²More information on <https://manifesto-project.wzb.eu/information/documents/corpus>

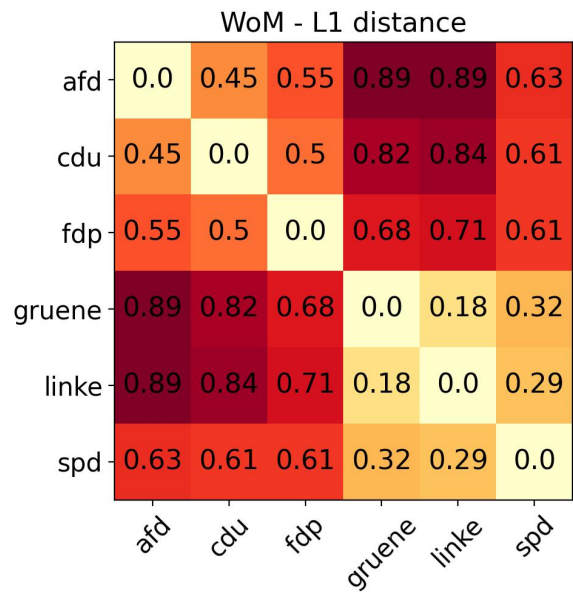
3.2 Ground Truth: Wahl-o-Mat

A problem with the task of predicting party proximity is to find a suitable ground truth against which to evaluate the models. In this study, we make use of a highly structured dataset, Wahl-o-Mat (WoM) from which we can construct a ground truth of party similarities with minimal manual involvement.

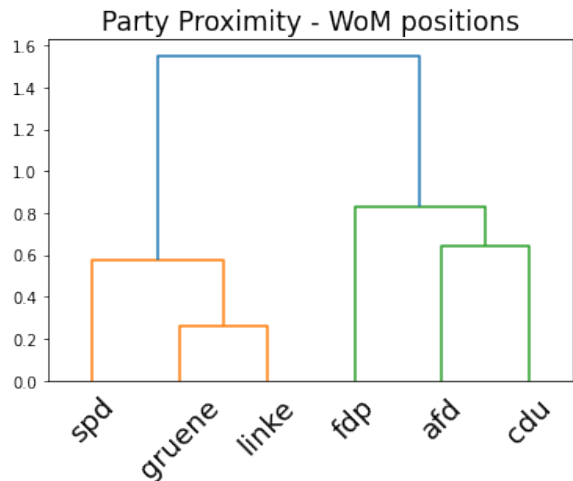
Wahl-o-Mat (WoM, [Wagner and Ruusuvirta \(2012\)](#)) is an online application that provides voting advice. The application collects users’ stances on a range of policy issues via a questionnaire. There are 38 issues in total and they cover a wide range of topics, e.g. ‘Germany should increase its defense spending’ or ‘The promotion of wind energy is to be terminated’. The users’ stances are then matched against those of the German parties in order to suggest the closest choices for users. The database behind WoM consists of the stances that each party takes towards each policy issue, which can be ‘agree’, ‘disagree’, or ‘neutral’.

WoM provides each user with a “percentage overlap” that they have with the different parties, suggesting that the set of policy issues and the stances are an informative basis for computing positional similarity ([Wagner and Ruusuvirta, 2012](#)). In this spirit, we define as our ground truth the *party distance matrix* which we obtain by representing each party by its vector of stances (represented -1, 1, 0) towards the different policy issues and computing the Hamming (L1) distances among them. Such distance calculations are used by political scientists to understand the overall (dis)similarity between party and voters ([McGregor, 2013](#)).

Figure 1a shows the distance matrix between parties: the higher the distance, the more they disagree on WoM policy issues. Figure 1b visualizes the ground truth differently, as an agglomerative clustering of the distance matrix. This ground truth arguably stands up to scrutiny: The two most left-oriented parties, Grüne (greens) and Linke (left), are most similar (distance 0.18), due to their similar environmental programs and shared concern about foreign policy. They are then most similar to social democratic SPD. On the other main branch of the clustering tree, which covers the right-oriented parties, AFD (right wing) and CDU/CSU (center conservative) are most similar, although less than the left parties (distance 0.45). Finally, the liberal party FDP groups with the conservative parties, but reluctantly so: it assumes a kind of bridge position between the left and right oriented parties.



(a) Distances between parties



(b) Agglomerative clustering

Figure 1: Based on Wahl-o-Mat policy positions.

4 Methods

We describe our method in three steps: (a) we define a set of informative text representations models; (b) we compute party similarities, parallel to Section 3.2, on the basis of these text representations; (c) we post-process the data.

4.1 Building Informative Text Representations

The first step is to build text representations that are informative for party similarity. As sketched above, we use neural language models (NLMs) as the current state of the art. This involves selecting a base embedding model and defining the different fine-tuning schemes.

Base embedding model: SBERT. We choose SBERT as the basis for our models. With its focus on sentence similarity and its computational efficiency, it is arguably the most appropriate model for our goals. Pre-trained SBERT without any fine-tuning³ serves directly as our first model.

Fine-tuning SBERT. Fine-tuning of SBERT can take place in different ways, but given our type of data, we use the triplet objective function where the model receives as input an anchor sentence a , a positive sentence p that is similar to the anchor sentence and a negative sentence n unrelated to both previous sentences. The objective of the fine-tuning is to minimize

$$\max(\|S_a - S_p\| - \|S_a - S_n\| + \epsilon, 0) \quad (1)$$

which encourages the model to learn that S_p is at least ϵ closer to S_a than to S_n . $\|\cdot\|$ is the distance metric, which is kept as the default Euclidean⁴. We experiment with two ways of constructing triplets for fine-tuning, first by *domain* and then by *party*.

SBERT_{domain} follows the same logic as in Dor et al. (2018) with the Wikipedia sections (and replicated in Reimers and Gurevych (2019)). We use the domain information from the manifestos (cf. Section 3) to construct triplets: The anchor and the positive sentences are part of the same domain and the negative sentence is from a different domain across party manifestos. The hypothesis is that aligning sentences by topic should help the model focus on relevant policy distinctions across parties.

SBERT_{party}, in contrast, intends to learn the distinction between the way parties express their claims or their ideologies and opinion. Here, we construct triplets by combining anchor sentences with positive sentences from the same party – irrespective of the domain – and negative sentences from the other parties’ manifestos. The hypothesis of this setup is that the embeddings incorporate the parties’ stances along with the way that particular sentences are presented, or styles used. We assume that many aspects of the text contribute to capturing the stance such as sentiment, text style and word usage.

³Pre-trained model: <https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2>

⁴Loss function and more details on: https://www.sbert.net/docs/package_reference/losses.html#sentence_transformers.losses.BatchAllTripletLoss

ID	Grouping	Filtering	Infor.
CLAIMDOM	Domain	Claims only	++++
CLAIM	-	Claims only	+++
DOM	Domain	All sentences	++
NONE	-	All sentences	+

Table 2: Models for the computational of party similarity, varying in the amount of information used

4.2 Four Models for Party Similarities

With the methods described in the previous subsection, we can obtain representations for individual sentences. We now need to define how to *aggregate* these sentences into global party representations – or rather, their similarities.

Table 2 shows four aggregating strategies that differ in the amount of information that they take into account. They differ in two main dimensions: (a), the *grouping*: is the similarity computed globally, over the complete manifestos, or domain by domain (b), the *filtering*: is the similarity based on all sentences in the manifestos, or only on sentences that contain concrete claims (cf. Section 1).

Regarding grouping, we hypothesize that it is easier for language models to assess the proximity between parties if sentences from matching topics are compared. Similarly, we expect that filtering by claims serves to focus the models on the ‘core’ of the parties’ policies.

CLAIMDOM: using claims and domains. In this, the most informed, model, we represent parties by the claims that they make, compare these claims by domain, and then average the by-domain similarities. Formally, let \vec{s} be the embedding produced for a sentence by an (implicit) encoder model, $cl(T)$ the set of claim sentences contained a text T , and $dom(P, i)$ the set of sentences for domain i in the manifesto of a party P . Then we can define the representation of a domain (Equation 1), the similarity for domain i (Equation 2), and a global similarity (Equation 3):

$$\vec{dom}(P, i) = \sum_{s \in cl(dom(P, i))} \vec{s} \quad (2)$$

$$\text{sim}(P_1, P_2, i) = \cos(\vec{dom}(P_1, i), \vec{dom}(P_2, i)) \quad (3)$$

$$\text{sim}(P_1, P_2) = \frac{1}{|Dom|} \sum_i \text{sim}(P_1, P_2, i) \quad (4)$$

CLAIM: using claims, but no domains. To compute similarities without domain information, we

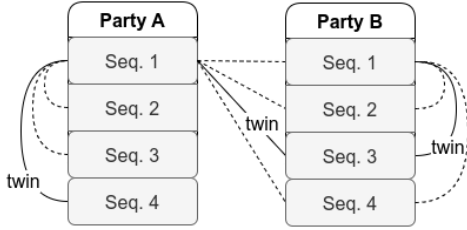


Figure 2: Twin matching: Solid lines mean pairings of maximal similarity.

could simply average over all sentences of the manifestos. However, pilot experiments showed that this procedure resulted in a severe loss of information. To avoid this, we introduce a method called *twin matching*, visualized in Figure 2. Twin matching maps each sentence in one manifesto to its nearest neighbor in the other manifesto (Equation 5) – in most cases, this will be a sentence of the same domain. Furthermore, we normalize the similarity to the twin by dividing by the maximum inter-claim similarity to both manifestos, and average over all sentences in the manifesto (Equation 7). Our hypothesis is that this procedure provides an approximating to domain-based grouping without the need for explicit domain labeling.

Formally, let $tw(s, T)$ denote the nearest neighbor, or twin, of sentence s in text T :

$$tw(s, T) = \arg \max_{t \in T} \cos(s, t) \quad (5)$$

Then the maximum inter-claim similarity C of a manifesto P , is

$$C(P) = \max_{p, p' \in cl(P) \wedge p \neq p'} \cos(p, p') \quad (6)$$

Then the similarity of two texts is:

$$\text{sim}(P_1, P_2) = \sum_{s \in cl(P_1)} \frac{\cos(s, tw(s, P_2))}{|cl(P_1)|(C(P_1) + C(P_2))} \quad (7)$$

DOM: using domains, but no claims. This model is identical to CLAIMDOM, but uses all sentences instead of just claims in Equation (2).

NONE: using neither domains nor claims. This model is identical to CLAIM, but uses all sentences instead of just claims in Equations (6) and (7).

4.3 Post-processing

As mentioned in Section 2, sequence representations should form an isotropic space for good similarity prediction. Therefore, we also experiment

with post-processed embeddings of the sentences by applying whitening transformation to our embeddings as suggested in Su et al. (2021). Following their normalization procedure, we start with a matrix $\mathbb{R}^{n \times d}$ representing n sequence vectors from a given encoding model with dimension d .⁵ Then, matrix W ($\mathbb{R}^{d \times d}$) is computed through singular value decomposition (SVD) and saved along with the mean vector μ ($\mathbb{R}^{1 \times d}$) retrieved from the initial input embedding matrix. Finally, every vector (\tilde{x}_i) of interest for the analysis is converted into our final representation as in $\tilde{x}_i = (x_i - \mu)W$.

Su et al. (2021) compute W and μ either with the data from the task at hand (train, validation and test set) or with data from another NLI task. In this study, we experiment with the same data of the analysis, i.e., the entire MaClaim21 in the CLAIMDOM and CLAIM models and Manifesto21 in the DOM and NONE models. This means that each sequence representation of the dataset is stacked into a matrix for the computation of W and μ .

5 Experimental Setup

5.1 Datasets

Fine tuning. We use the German Manifesto data for 2013 and 2017 to fine-tune SBERT following Section 4.1. There is a deliberate temporal gap between the fine tuning datasets and the year of our ground truth, namely 2021, to ensure that the model picks up generalizable differences between parties rather than overfitting. However, we acknowledge the drawback that fine-tuning does not receive any signal from newly emerged topics (e.g. Covid19) and that party communication has not transformed drastically over the last four years.

Appendix A.3 provides more details and statistics, including evaluation on a 20% held-out validation set, which shows that fine-tuning improves both $SBERT_{party}$ and $SBERT_{domain}$ over plain SBERT, with $SBERT_{domain}$ gaining most.

Party representation. To compute party similarities following Section 4.2, we use the 2021 manifestos, which arguably form the right textual basis to evaluate against our Wahl-o-Mat ground truth for the 2021 German elections (Section 3.2). Recall that the Manifesto data comes with annotated domains, but not with annotated claims. We therefore applied an automatic claim classifier to identify claims (Blokker et al., 2020). We evaluated the

⁵The pre-trained model we use has 768 dimensions.

Model + postproc.	MaClaim21		Manifesto21	
	CLAIMDOM	CLAIM	DOM	NONE
	(++++)	(+++)	(++)	(+)
fasttext _{avg}	0.17	0.30	0.27	0.28
fasttext _{avg} +whiten	0.54*	0.35	0.44*	0.41
BERT _{german}	0.12	0.28	0.11	0.27
BERT _{german} +whiten	0.37	0.47*	0.36	0.48*
RoBERTa _{xml}	0.03	0.35	0.08	0.33
RoBERTa _{xml} +whiten	0.39	0.51*	0.46*	0.54*
SBERT	0.38	0.47*	0.31	0.47*
SBERT(whiten)	0.57*	0.50*	0.53*	0.57*
SBERT _{domain}	0.22	0.23	0.32	0.16
SBERT _{domain} +whiten	0.44*	0.45*	0.41	0.52*
SBERT _{party}	0.45	0.13	0.32	0.16
SBERT _{party} +whiten	0.53*	0.70*	0.50*	0.69*

Table 3: Experimental results: Mantel’s correlation between categorical and textual distance matrices. +whiten means that the models have undergone whitening postprocessing. The + symbol indicates the level of informativeness from Table 2. Highest correlation for each model in boldface. * p-value < 0.05.

results of the classifier by calculating the precision on a subset of 324 manually labeled claims from the 2021 manifestos and obtained a reasonable precision of 75,6%. More information about data and classifier can be found in Appendix C.1.

This procedure results in two datasets for model training: Manifesto21 (with domain annotation) has 17,052 sentences; MaClaim21 (with domain and claim annotation) consists of 9,814 claims. More details and statistics are in Appendix B.

5.2 Models

In our empirical evaluation below, we vary the following three parameters: (1), Embedding model and fine-tuning (SBERT plain vs. SBERT_{domain} vs. SBERT_{party}). (2), Party similarity computation (CLAIMDOM vs. CLAIM vs. DOM vs. NONE). (3), Postprocessing (whitening vs. none). We consider all combinations of these parameters.

Baselines We consider three baselines. The first and simplest one is a pre-trained FastText model for German based on character n -gram embeddings (Bojanowski et al., 2017). We compute sentence representations by tokenizing the sentences based on the FastText tokenizer and averaging all FastText token representations.⁶

⁶We evaluated both on the general version of fasttext for German available on fasttext.cc and also on a trained version with newspaper articles from TAZ for a more domain specific model. Since both models obtained comparable results, we report only results for the former.

The other two baselines use transformer-driven (sub)word embeddings, namely from BERT-German⁷ and multilingual RoBERTa-XLM⁸. We choose the former because monolingual models often perform better than multilingual ones and the latter because it is the student model with which SBERT has been trained, which allows us to check how much better SBERT can be in a text similarity task in the political domain. Again, we feed each sentence to these models and compute the final representations by averaging all token representations from the two last layers of the model, a strong baseline for similarity tasks (Li et al., 2020; Su et al., 2021).

5.3 Evaluation

To evaluate the pairwise party similarities computed by the models, we turn them into distances and compare them against our ground truth distance matrix (Section 3.2) with the Mantel test (Mantel, 1967). This test is a variant of standard correlation tests (such as Spearman’s ρ) which are not applicable to distance matrices because they assume that the observations are independent of one another. In our case, changing the position of one value in the matrix would change the correlation between a pair or parties. Having said that, the Mantel test addresses this problem by calculating correlations on

⁷<https://huggingface.co/bert-base-german-cased>

⁸<https://huggingface.co/xlm-roberta-base>

all permutations of the flattened distance matrix. The two-tail hypothesis tests whether the correlation between the ground truth matrix and the target distance matrix is statistically significant or not. We use the nonparametric version of the test since the party distances are not normally distributed.

6 Results and Discussion

Table 3 shows the quantitative results of our experiments. We first discuss the effect of our various experimental parameters.

Effect of postprocessing. By comparing the upper and the lower row in each colored block, we observe that the whitening transformation is beneficial in nearly all models, and where it is not, the loss is minor. On average, post-processed model embeddings are 22 percentage points higher in the correlations, and consistently obtain significant correlations with the ground truth. This suggests that the benefit of enforcing isotropic distributions extends to the domain and genre of political texts. Given the substantially higher performance of the models with the post-processing step, we focus on their results for the remainder of this discussion.

Effect of embedding models and fine-tuning. Comparing the rows in the table, we observe that our two baseline models, BERT and RoBERTa, show generally worse performance than even the non fine-tuned SBERT. BERT is generally the worst performer among the three, despite its monolinguality, which we interpret as evidence that the architectures more geared towards similarity tasks have an advantage. We take these results as validation of our choice of SBERT as embedding model.

Interestingly, our simplest baseline, *fasttext_{avg}*, performs better than most models in the most informative scenario (Mantel=0.54) and relatively well with domain information (Mantel=0.44), but degrades when less information is available. This suggests that FastText embeddings are informative enough to support generalization from rich annotation, but are not able to align semantically similar sentences well in a less informative scenario such as in the twin matching approach.

Among the fine-tuned variants of SBERT, *SBERT_{domain}* performs surprisingly badly and is generally outperformed by vanilla RoBERTa. This suggests that optimizing the model to pick up on domain contrasts is distracting the model from capturing the dis(similarity) between parties.

In contrast, *SBERT_{party}* does very well, and competes with vanilla SBERT for the best results. Indeed, SBERT wins in both setups that are grouped by the domain category (CLAIMDOM and DOM), reaching 0.57 and 0.53, respectively. Conversely, *SBERT_{party}* wins the two scenarios without the grouping by domains (CLAIM’s Mantel=0.70 and NONE’s Mantel=0.69), and achieves the overall highest correlations here.

These results suggest that SBERT, without any fine-tuning, is reasonably good at capturing the proximity between parties if more information is provided: if we have both only claim structure and the domain category then SBERT can be enough (Mantel=0.57). If there is unstructured data, but there is still domain information, despite having a drop in performance, it can still achieve a reasonable correlation (Mantel=0.53).

SBERT_{party}, in contrast, performs better in the settings without domain information, that is, when the party similarity is based on twin sentence similarity (Section 4.2). We believe that this is the case because the sentence-level fine-tuning of *SBERT_{party}* is most directly carried forward into the predictions of the model. In effect, therefore, fine-tuning SBERT by contrasting the party difference is the best way to encode fine-grained differences between parties’ views and ideologies.

Analysis by agglomerative clustering. To complement the analysis by correlation coefficients in Table 3, we compute agglomerative clusterings with average linkage for the best models from Table 3. The results, shown in Figure 3, show a good correspondence to the quantitative results, thus lending support our use of the Mantel test.

Indeed, the two SBERT models in 3(a) and 3(c), which reach moderate correlation coefficients, disagree substantially with the ground truth clustering: they group, for example, the far right AFD with the liberal FDP in (a), and with the left wing Linke in (c). Also, the conservative CDU is grouped with Grüne (greens) and social democratic SPD. In contrast, the two *SBERT_{party}* models in 3(b) and 3(d) show a better match with the ground truth, even though both group Grüne with SPD instead of Linke, and (b) has AFD as an outlier altogether.

General outcome. Probably the most striking outcome of our experiment is that the best results – both in terms of the correlation coefficient and in terms of the clustering – results from models

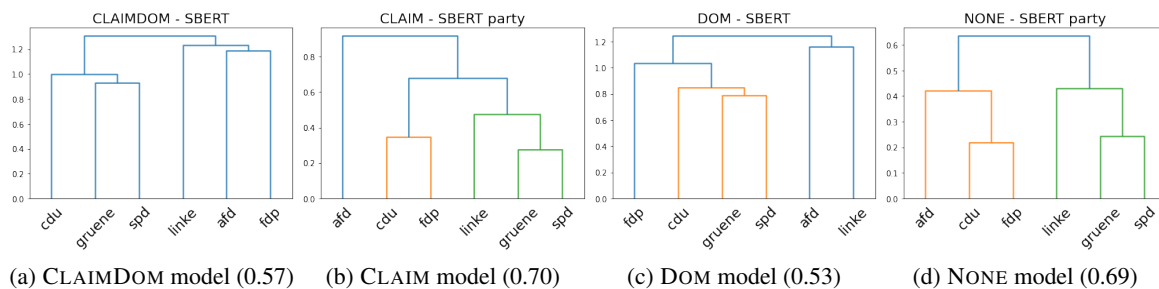


Figure 3: Agglomerative clustering for the best model of each setting. Mantel correlation in parenthesis. Ground truth’s comparison in Fig. 1b.

that use very little structured information (CLAIM, NONE). The difference among the two is small, and can be seen as a trade-off between using a larger, more noisy dataset (all sentences: Manifesto21) and a more focused dataset (just the claims: MaClaim21) of about half the size. These results confirm the idea that it is possible to use natural language processing methods to identify the dis(similarity) between party according to their policy positions with unstructured data.

We believe that this result is a combination of a good choice of fine-tuning regimen – providing the embeddings with a signal concerning the contrast between parties – with an appropriate way to model similarity, with our twin matching approach which helps to match the most relevant parts of the two manifestos to one another. These two aspects reinforce each other, since a well fine-tuned model is better able to push away dissimilar parties while bringing closer together similar ones.

7 Conclusion

In this paper, we have investigated to what degree text representations can capture the proximity of parties and how to best fine-tune representations for this task. Our results indicate that aspects that have been proposed as important for this type of analysis in political science, namely annotation of domains (Burst et al., 2021) and claims (Koopmans and Statham, 1999), do not appear to matter greatly for this task – or at least, manual annotation can be replaced by NLP tools: we have recognized claims with a classifier (Blokker et al., 2020) and have proposed a weekly supervised method, “twin matching”, to approximate domain-level similarity computation. Indeed, one of our models that does not use any manual annotation is among the top contenders. Of rather greater importance for party similarity prediction, according to our findings, is

fine-tuning the text representations and post-processing them.

This is good news for computational political science: the judicious use of document structure appears able to help alleviate the effort of having domain experts annotate large corpora. The two main limitations of our current study relate to this outlook: (a) we only experimented with a single language and ground truth – future work should take into account multiple languages and time periods, with a potential long term goal of text-based models for party development (König et al., 2013); (b) we only scratched the surface of cues available for fine-tuning. Future work could, for example, take into account other aspects of parties such as ideological position (Glavaš et al., 2017), or reach beyond manifestos to include information from other types of party interactions (Strom, 1990). In addition to that, work on interpreting both the fine-tuned and vanilla SBERT models would be interesting to better understand the predominant dimensions of the sentence representations in the political domain.

Acknowledgments

We acknowledge funding by Deutsche Forschungsgemeinschaft (DFG) for project MARDY 2 (375875969) within the priority program RATIO.

Ethics Statement

We believe that this study does not carry major ethical implications in terms of data privacy or handling, given that our datasets are based on publicly available party manifestos from the German elections and from a public and freely accessible voting advice application (Wahl-o-Mat). The annotators that provided us with a subset of labeled claims to estimate the quality of the claim classifier were

student assistants from the university remunerated fairly according to their working hours.

References

- Nico Blokker, Erenay Dayanik, Gabriella Lapesa, and Sebastian Padó. 2020. [Swimming with the tide? positional claim detection across political text types](#). In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 24–34, Online. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Ian Budge. 2003. Validating the manifesto research group approach: theoretical assumptions and empirical confirmations. In *Estimating the policy position of political actors*, pages 70–85. Routledge.
- Ian Budge, Hans-Dieter Klingemann, Andrea Volkens, Judith Bara, and Eric Tanenbaum, editors. 2001. *Mapping Policy Preferences: Estimates for Parties, Electors, and Governments 1945-1998*. Oxford University Press, Oxford, New York.
- Tobias Burst, Werner Krause, Pola Lehmann, Jirka Lewandowski, Theres Matthieß, Nicolas Merz, Sven Regel, and Lisa Zehnter. 2021. Manifesto corpus. version: 2021.1. *Berlin: WZB Berlin Social Science Center*.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. [Universal sentence encoder for English](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. 2018. [Learning thematic similarity metric from article sections using triplet networks](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Melbourne, Australia. Association for Computational Linguistics.
- James N Druckman, Lanny W Martin, and Michael F Thies. 2005. Influence without confidence: Upper chambers and government formation. *Legislative Studies Quarterly*, 30(4):529–548.
- Kawin Ethayarajh. 2019. [How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tiejun Liu. 2019. [Representation degeneration problem in training natural language generation models](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. [Unsupervised cross-lingual scaling of political texts](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 688–693, Valencia, Spain. Association for Computational Linguistics.
- Ruud Koopmans and Paul Statham. 1999. Political claims analysis: Integrating protest event and political discourse approaches. *Mobilization: an international quarterly*, 4(2):203–221.
- Thomas König, Moritz Marbach, and Moritz Os-nabrügge. 2013. [Estimating party positions across countries and time—a dynamic latent variable model for manifesto data](#). *Political Analysis*, 21(4):468–491.
- Gabriella Lapesa, André Blessing, Nico Blokker, Erenay Dayanik, Sebastian Haunss, Jonas Kuhn, and

- Sebastian Padó. 2020. DEbateNet-mig15: tracing the 2015 immigration debate in germany over time. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 919–927.
- Michael Laver, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American political science review*, 97(2):311–331.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. [On the sentence embeddings from pre-trained language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [RoBERTa: A robustly optimized BERT pretraining approach](#). *ArXiv*, abs/1907.11692.
- Nathan Mantel. 1967. The detection of disease clustering and a generalized regression approach. *Cancer research*, 27(2):209–220.
- R Michael McGregor. 2013. Measuring “correct voting” using comparative manifestos project data. *Journal of Elections, Public Opinion and Parties*, 23(1):1–26.
- Federico Nanni, Goran Glavaš, Ines Rehbein, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2022. Political text scaling meets computational semantics. *ACM/IMS Transactions on Data Science (TDS)*, 2(4):1–27.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of EMNLP/IJCNLP*, pages 3980–3990. Association for Computational Linguistics.
- Jonathan B Slapin and Sven-Oliver Proksch. 2008. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science*, 52(3):705–722.
- Kaare Strom. 1990. [A behavioral theory of competitive political parties](#). *American Journal of Political Science*, 34(2):565–598.
- Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. [Whitening sentence representations for better semantics and faster retrieval](#). *ArXiv*, abs/2103.15316.
- Markus Wagner and Outi Ruusuvirta. 2012. [Matching voters to parties: Voting advice applications and models of party choice](#). *Acta Politica*, 47(4):400–422.

A Appendix

A.1 Fine-tuning data

Party	Num. inst.
Grüne	5913
Die Linke	4243
Social Democratic Party of Germany (SPD)	3566
Free Democratic Party (FDP)	3149
Christian Democratic Union (CDU)	2569
Alternative for Germany (AfD)	770

Table 1: Number of instances in the train set of the fine-tuning of SBERT_{party}. Data from the 2013 and 2017 manifestos.

Domain name	Num. inst
Welfare and Quality of Life	7078
Economy	6330
Fabric of Society	2586
Freedom and Democracy	2395
External Relations	2306
Social Groups	2144
Political System	1682

Table 2: Number of instances in the train set of the fine-tuning of SBERT_{domain}. Data from the 2013 and 2017 manifestos. More information about the categories can be found on https://manifesto-project.wzb.eu/coding_schemes/mp_v5

Party	Year	Sentence	Domain
AfD	2017	This oligarchy holds the levers of state power, political education and informational and media influence over the population.	Political System
CDU	2017	We have set ourselves an ambitious goal: We want full employment for all of Germany by 2025 at the latest.	Social Groups
FDP	2013	We want to continue to give people the freedom to pursue their ideas - creating growth, progress and prosperity for all.	Freedom and Democracy
Grüne	2013	We want to make a change today to move towards an economy that benefits everyone, not just a few.	Welfare and Quality of Life
Die Linke	2013	But the populations and workers of these countries have common interests: the fight against wage depression, recession and mass unemployment.	Economy
SPD	2017	This includes ensuring that social cohesion in our country becomes stronger again and that decent dealings with one another are not lost to political radicalization.	Fabric of Society

Table 3: Examples from the training dataset with their corresponding domain names translated from German.

A.2 S-BERT training parameters

- Pre-trained model: paraphrase-multilingual-mpnet-base-v2
- Maximum sequence length: 128
- Train batch size: 16
- Number of training epochs: 5
- Learning rate: 2e-5
- Warm up steps: 100

A.3 Fine-tuning evaluation

Model	f1	SBERT (f1)
SBERT _{domain}	71,39%	66,66%
SBERT _{party}	68,79%	66,66%

Table 4: Comparison of the f1 scores between the non-fine-tuned and fine-tuned SBERT models on the held out validation set.

B Appendix

B.1 Data for the evaluation

Party	Num. claims
Die Linke	2770
Gruene	2380
CDU	1685
FDP	1388
SPD	952
AfD	638

Table 5: Number of claims per party in MaClaim21.

Party	Num. sentences
Die Linke	4850
Gruene	3947
CDU	2775
FDP	2239
SPD	1665
AfD	1574

Table 6: Number of sentences per party in Manifesto21.

C Appendix

C.1 Claim identifier

The claim identifier was trained on annotated data from the DebateNet dataset (Lapesa et al., 2020). The annotations are based on news articles from the German newspaper TAZ regarding the migration in the domestic scenario. Sentences that contain a claim are considered as positive and sentences without any claims are negative. It has been verified that the claim identifier trained on DebateNet can transfer reasonably well to the party manifestos (Blokker et al., 2020) with an averaged f1 score of 82% across the election campaigns of 2013 and 2017. More information regarding the training process:

- Number of training instances: 13,283
- Number of validation instances: 1,477
- Number of testing instances: 1,641
- Maximum sequence length: 128
- Train batch size: 32
- Number of training epochs: 5
- Learning rate: 3e-5

C.2 Evaluation on 2021 party manifestos

Expert annotators from the political science faculty annotated 324 unique political claims from six major German parties competing in the federal election of 2021. Annotations of claims followed a fine-grained hierarchical ontology (*codebook*) yielding 75 unique sub-categories that are divided into eight major categories. While the latter broadly corresponds to relevant policy fields, such as ‘health’, ‘economy and finance’, or ‘education’, the former specifies the concrete policy measure to be taken, for instance, ‘mandatory vaccination’, ‘raise taxes’, ‘expansion of education and care services’. We do not provide the inter-annotator agreement because annotators worked closely together in this task. However, we verified the quality of the dataset by having a third annotator gold standardizing the dataset.

The classifier detected 245 out of the 324 annotated claims, reaching a reasonable precision of 75,6%. In total, the classifier predicted 9,814 claims out of 17,052 sentences.