Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bachelor Thesis

Topic Modeling and Stance Detection for Accessible Legislative Sessions

in the Language Technology (LT) group

by

Anton Gustav Trappe
born on 14.09.2001
Matriculation number: 7501564
Field of study: B. Sc. Informatik

submitted March 10, 2025

Supervisor: Tim Fischer
First Reviewer: Prof. Dr. Chris Biemann
Second Reviewer: Tim Fischer

Contents

1 Introduction

Germany is a parliamentary democracy. There are many parliaments in Germany at the different levels of the Federation. Most prominently there is the Bundestag, the parliament at the federal level. In the 20th election period alone there were 212 Sessions with almost 25,000 speeches totaling over 1500 hours of sessions in the Bundestag. From those numbers, it becomes clear that it is not viable to watch all the sessions of the Bundestag just to stay informed. Even most of the members of the parliament (MPs) themselves are only present for the, to them, important sessions; so even they do not hear everything said in the Bundestag. Currently, people inform themselves over two information channels about the Bundestag sessions. Traditional news outlets are covering the Bundestag, and there are social media where clips from Bundestag sessions circulate. Especially the social media clips are heavily curated sometimes by political actors with ulterior motives. But while the Bundestag receives comparatively large coverage, it is by far not the only German parliament. There are 16 state parliaments and about 400 local parliaments. Those smaller parliaments get even less news coverage. For those reasons the exploration of ways to summarize legislative sessions is relevant.

1.1 Prior study

I already published a web application that summarizes the sessions from the German Bundestag [1]. The application uses the official transcripts from all Bundestags sessions as basis and visualizes the information found in them in multiple ways. The goal is to show the users machine generated insight into speeches, sessions and MPs, while always allowing them to read the speeches themself. To do this multiple features were implemented that are described below.

The user can look at each session from the 20th election cycle. First, a summary of the session is shown (Figure 1.1). There the user can see how long the session lasted. To set this in relation to the other sessions a histogram is shown comparing all session durations with each other. The bucket the currently selected session falls in is marked red. The second histogram presented to the user shows how often heckling occurred in the selected session compared to all other sessions. Below those histograms the user can see a Wordcloud showing the words that were used the most in the selected session. The colors indicate which party used the word the most, relative to their number of seats. If the word would not be colored relative to the number of seats, most words would be colored in the color of the biggest parties.

Each session in the Bundestag is organized into agenda items by the MPs. Those agenda items are listed below the initial session summary. For each agenda item a Wordcloud created from all speeches given under that agenda item is shown. The user can click on each agenda item and reach a new summary page for the clicked agenda item. This agenda item overview first shows the Wordcloud already listed on the session page for the agenda item but larger. This Wordcloud is usually more expressive than the Worcloud for the whole session because the shown words are

1. https://basecamp-demos.informatik.uni-hamburg.de/bundestagsanalysen/ accessed March 9, 2025

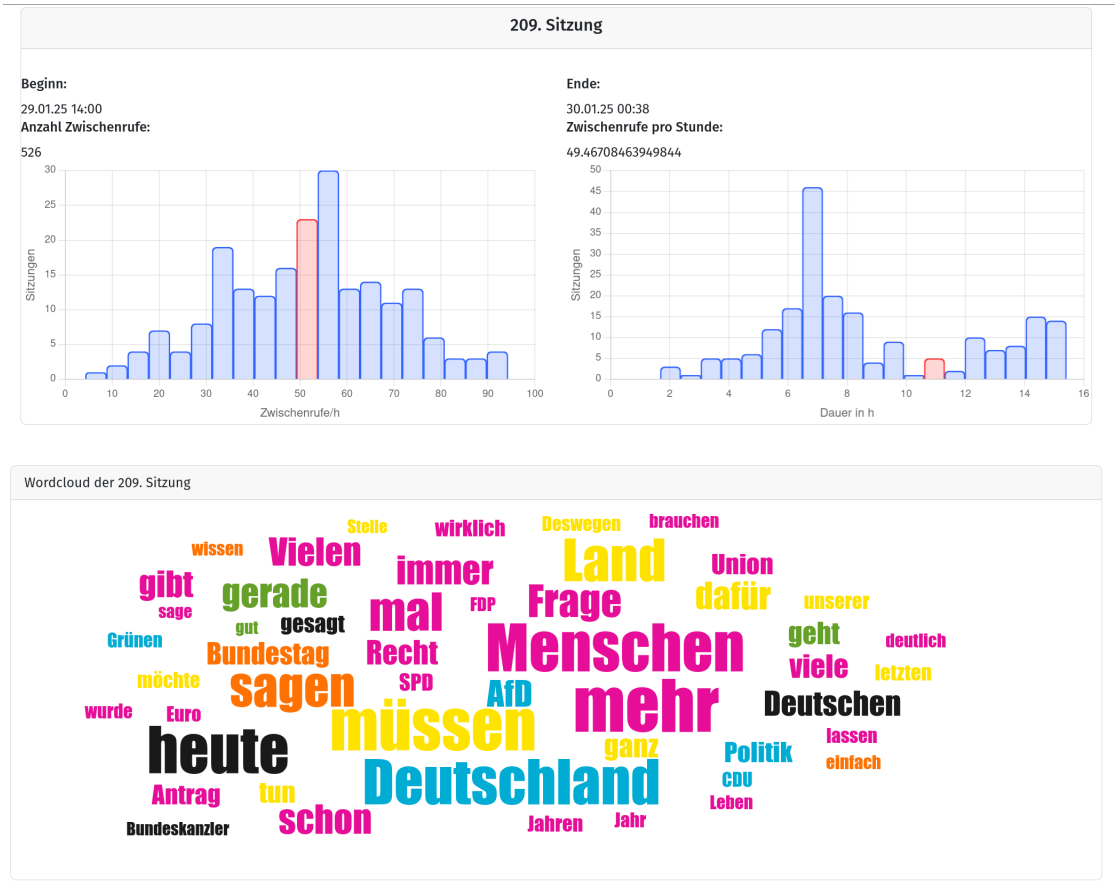

Figure 1.1: The session summary view. It shows a histogram that visualizes how often heckling occurs per our and a histogram comparing the duration of this session to the others. Below that a Wordcloud compiled from all speeches in the 209th session is shown.

more coherent. This is because one agenda item usually is about the same topic while one whole session usually consists of several unrelated agenda items. Below this, all speakers, who gave a speech on that agenda item are listed. Figure 1.2 shows the beginning of such a list. A histogram comparing the length of the speech to the length of all other speeches is shown on the left side. In this example all speeches are pretty long because they were given by high profile politicians. Right of this histogram another Wordcloud is shown, compiled from the Words in the single speech. The colors still convey the same meaning and are always calculated globally over all speeches from the election cycle. Although the goal of both the prior project and this work is to summarize the proceedings in the parliament by machine, it is important for transparency reasons to allow the user to see the underlying data, in this case the speeches. When the user clicks on the "*Volltext*" link they will get to a page where they can view the speech.

This speech page shown in Figure 1.3 shows three histograms to the user. First, the histogram about the length of the speech. Second, the amount of heckling that occurred in this speech compared to all other speeches. Lastly, the amount of applause the speaker received for this speech. This could be an indication of how well the speech was received and the amount of heckling can be an indication of how controversial the speech was, as heckling often is done by MPs with opposing views. But both of those metrics obviously correlate with the length of the speech. Also speakers with a high profile usually receive more applause and are targeted more by heckling. Therefore those interpretations should be done cautiously. The original speech is shown below those histograms. It is visualized in a chat like style, where the speakers text is shown in green chat bubbles. Information such as applause is shown in grey like system messages in a chat. Heckling or other expressions by other MPs is shown in blue (Figure 1.3). This layout was picked because most users are familiar with this chat like interface.

The user also can view a page about each MP that is reachable through a link whenever the name of the MP appears somewhere or through a search bar at the top of each page. This page is seen in Figure 1.4. On the top there are general information including which party the MP is part of and a short biography provided by the Bundestag. Below that there is a histogram comparing the number of speeches the MP has given to how many speeches other MPs gave. Next to that it is possible to get to individual speech pages given by the currently selected MP. Shown are the 10 most recent speeches, the 10 speeches where the most heckling occurred, the 10 speeches during which the MP received the most applause, and the 10 longest speeches given by the MP. Below a Wordcloud compiled from all speeches by that MP is shown below. Under this there are information about the heckling and the applause. This can be seen in Figure 1.5. First, a Histogram shows how often the MP heckled compared to the other MPs. The Wordcloud next to that shows the words the MP uses most often when they heckle. Below that there are two lists of other MPs. First, the five MPs during whose speeches the selected MP heckles the most are listed. Second, the five MPs, who heckle the most during speeches of the selected MP. Below that there is a Histogram comparing how often the selected MP receives applause to the other MPs and next to that is a bar chart that visualizes from which parties the MP receives the most applause.

Lastly, every time a Party is shown on any page the user can click on it and reach a summary page for that Party (Figure 1.6). There a Wordcloud over all speeches given by all members of that party is shown. Then, there are two bar charts visualizing from which parties speakers from the selected party receive the most applause and which parties' speaker the selected party applauded the most.

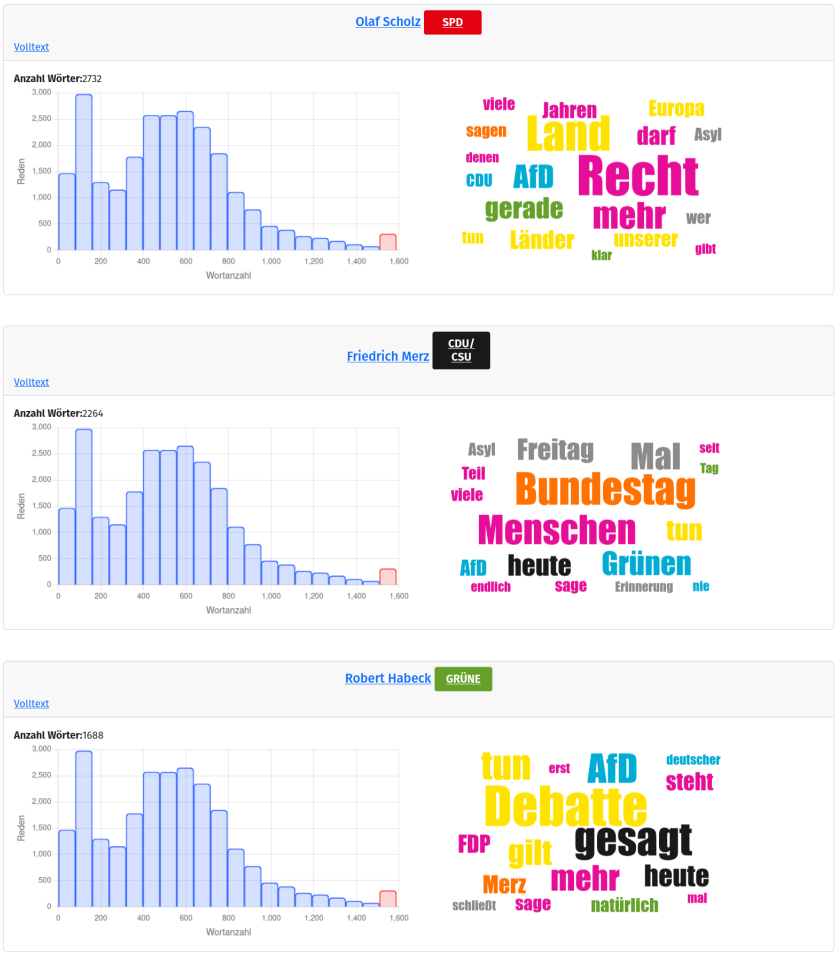

Figure 1.2: The begin of the speaker list for the first agenda item of the 209th session

Figure 1.3: The page for a single speech, making the original speech text available to the user.

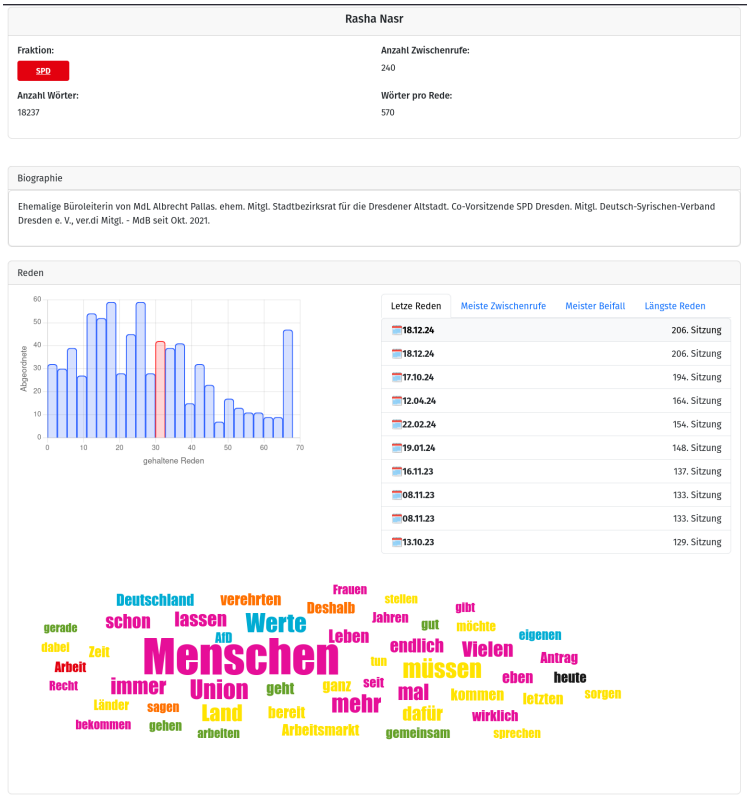

Figure 1.4: The overview page for the MP Rasha Nasr(SPD)

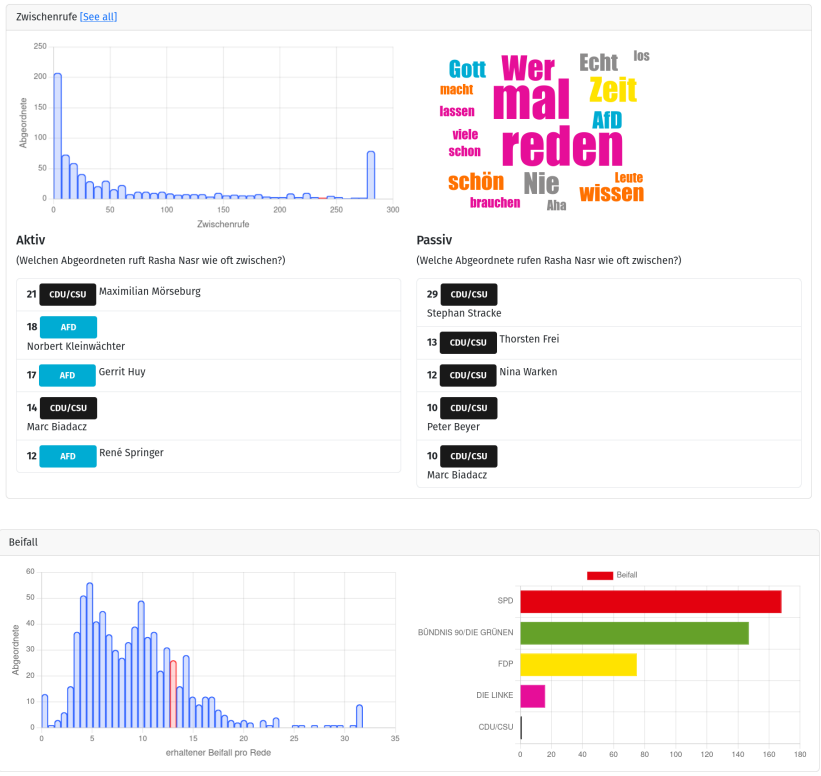

Figure 1.5: Aggregated heckling and applause information for the MP Rasha Nasr

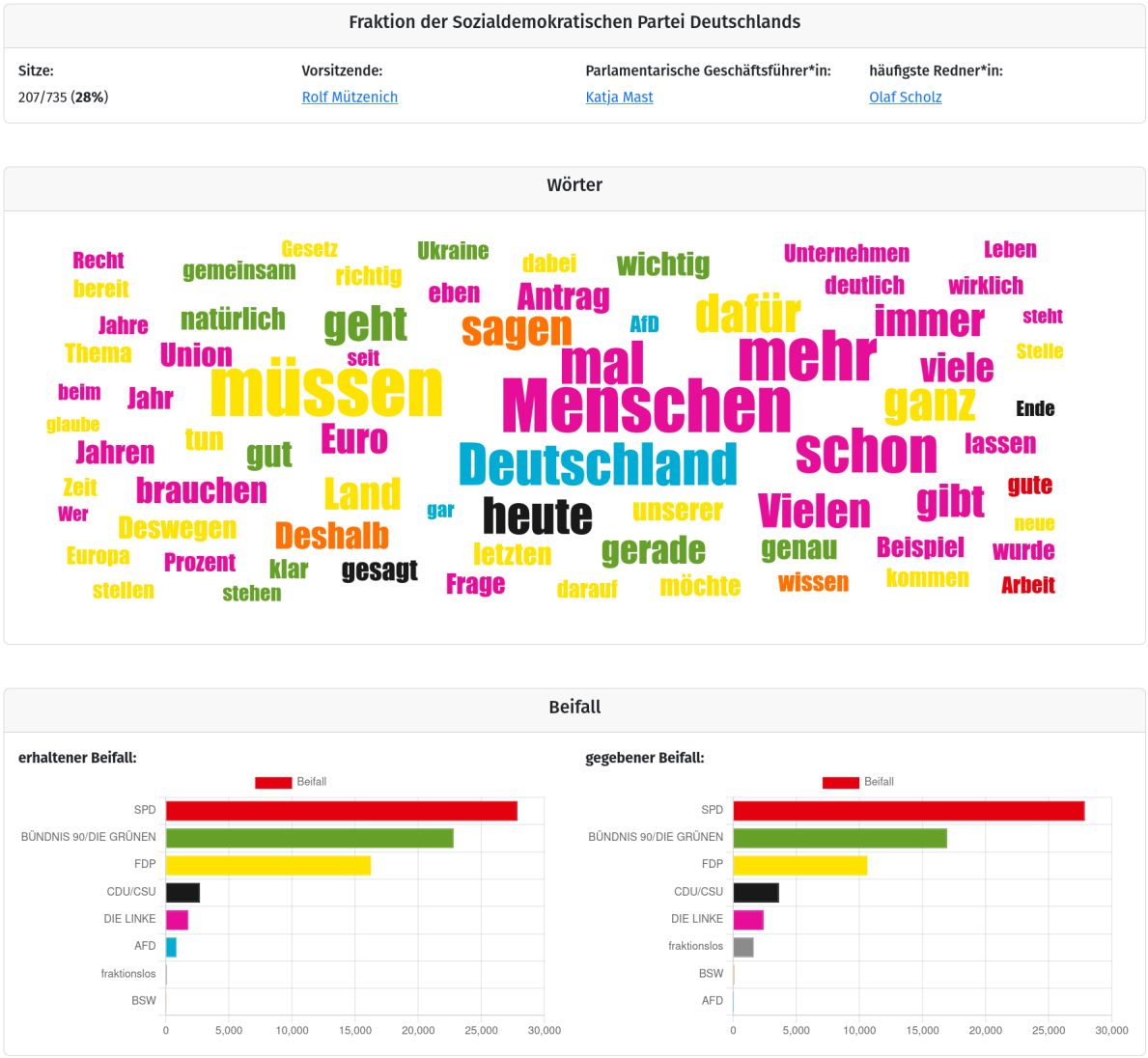

Figure 1.6: The party overview

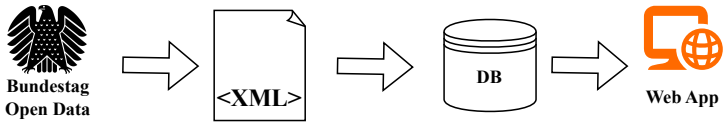

Figure 1.7: The technical side of the existing web app. First, the session protocols are scraped from the German Bundestag as files in the XML-format. Then, those XML files get parsed and the information from them is written into an PostgreSQL database. The Web app queries this database to visualize the information.

Figure 1.7 how the application works from a technical point of view. The data for the application is scraped from the Open Data service of the Bundestag [2]. There all session protocolls are available in XML format. The scraping is done using the Scrapy framework [3] The scraped XML-files are parse using the untangle library [4]. Then the parsed data is written into a PostgreSQL database using sqlalchemy [5]. Then fastapi[6] is used for the backend of the application and the frontend visualization is done with chart.js [7].

1.2 Motivation

The structure of the application in its current state follows the structure in which the Bundestag operates. It is structured by sessions and agenda items. This structure is not helpful to a user, who is interested in how the MPs discuss a certain topic, but has no knowledge of when this topic was discussed in the Bundestag. Orienting themself by the names of the agenda items is not simple because those names are formulated in highly technical language. They usually contain more information about the legislative process than about the topic. For those reasons grouping the speeches by discussed topic and finding appropriate labels for these topic groups, would help users to find relevant speeches.

Currently, a user interested in the position of the speaker can gain some insights about the controversiality of a speech based on the amount of heckling that occurred. But controversiality is for most users not the relevant criterion to form their opinion. Therefore, it would be valuable to the users if the position of actors in the Bundestag about the earlier extracted topics would be shown. For both extracting topics and assigning stances, there are existing methods. But which of those methods works best for the existing German parliamentary corpus needs to be evaluated, and while the methods are there, using them on german parliamentary debate is not very proliferated. For that reason finding ways to visualize the results of those methods is relevant. With those problems in mind the following research questions were formulated, with the goal of extending the current application through topic modeling and stance detection.

1.3 Research questions

RQ1: What is the best topic modeling method to extract meaningful topics from parliamentary sessions?
I plan to use topic modeling on the content of the speeches. More specifically I will compare the results from LDA (Blei et al., 2003), BERTopic (Grootendorst, 2022) and TopicGPT (Pham et al., 2024) to decide which model works better for what. Then, it will be possible to tell what topics were discussed and also what topics a specific MP covers.

RQ2: What stance detection method is most suited to find the positions of speakers on the topics extracted prior?
Especially in politics, it is not only interesting what topics were discussed but also how political

2. https://www.bundestag.de/services/opendata accessed March 9, 2025
3. https://scrapy.org/ accessed March 9, 2025
4. https://untangle.readthedocs.io/en/latest/ accessed March 9, 2025
5. https://www.sqlalchemy.org/ accessed March 9, 2025
6. https://fastapi.tiangolo.com/
7. https://www.chartjs.org/ accessed March 9, 2025

actors think about the topic, I will compare three different unsupervised approaches to stance detection: One classic approach under the Bag of Words Assumption called Wordfish (Slapin and Proksch, 2008) one approach using Word Embeddings and semantic similarity (Glavaš et al., 2017), and I will prompt a Large Language Model (LLM) to evaluate the Stance from the speech directly.

RQ3: How can the results of both the topic modeling and subsequent stance detection be presented visually to give useful insights into the speeches given in a parliament?
A good visualization should give useful information about the content of a session, or it should foster an understanding of how a speaker views certain topics. The visualization should be intuitively understandable in a short time, to serve as a fast way to gain an understanding of the proceedings in the parliament. While allowing for fast understanding it is important to allow the user to see the provenance of the analysis. In this case users should have a easy way to read the speeches on which the visualization is build on top of.

The following work is organized as follows: First, in Chapter 2 different topic modeling techniques are explored, then Chapter 3 does explores multiple stance detection approaches. Chapter 4, I will discuss the visualization of both topic modeling and stance detection and explain how this visualization was integrated into the existing web application. Lastly in Chapter 5, I will discuss possible improvements and open areas to explore adjacent to my application, and conclude this work.

2 Topic Modeling

2.1 Introduction

2.1.1 Motivation

Organizing documents by finding fitting topics for them is a very intuitive approach, to structure large corpora. We organize mails into folders, books into genres and news articles into resorts. All this can be considered assigning topics to those texts. To assign texts to topics obviously finding the topics themself is needed. This is what topic modeling is useful for: It aims to extract topics from a corpus and assign each document from the corpus to one or more of those topics. Finding the topics that define a text corpus can create a understanding of the corpus without the need to look extensively into the corpus.

Topic modeling is useful for understanding the speeches in the German Bundestag because it gives the large amount of speeches an easily understandable structure. This structure allows a user to navigate to the speeches they might be interested in more easily. Furthermore, topic modeling is an important basis for further analysis. It provides targets for stance detection. And it can serve for quantitative analysis of the frequency of certain topics in different time periods or between different parties.

2.1.2 Task Definition

Given a corpus of documents, topic modeling assigns each document d of the corpus a topic t. The topic t is chosen from a list that needs to be extracted from the corpus. Alternatively each document d can be assigned a distribution over all topics $(t_0, t_1, .., t_n)$ where t_i is the probability d is part of topic t_i or the portion to which d consists of t_i. But for this paper I assume each document consists only of one topic. Topic modeling is usually done unsupervised. Blei, 2012 defines topic modeling as follows: „topic models are algorithms for discovering the main themes that pervade a large and otherwise unstructured collection of documents. Topic models can organize the collection according to the discovered themes." And both Grootendorst, 2022 and Egger and Yu, 2022 note that the goal of topic modeling is to find hidden or latent topics: „Topic models can be useful tools to discover latent topics in collections of documents" „In a nutshell, a topic model is a form of statistical modeling used in machine learning and NLP, as discussed earlier that identifies hidden topical patterns within a collection of texts"

2.1.3 Goal

I want to find a topic modeling approach, that works well for the speeches from the German Bundestag. The approach should find coherent groups of speeches to assign to exactly one topic. Those topics should be represented in a understandable way for a user. The topics should give a structure to the corpus of the Bundestag speeches, and provide a basis for further analysis.

2.1.4 LDA

Latent Dirichlet allocation (LDA) is a topic modeling method first presented by Blei et al., 2003. The key assumption for LDA is that a document is generated from an underlying probability distribution. This generation process for a document can be described in the following way:

1. Choose the length of the document N
2. Choose $\theta \sim Dir(\alpha)$
3. For each of the N words:
    a) Choose a topic $z_n \sim Multinomial(\theta)$
    b) Choose a word w_n from $p(w_n|z_n, \beta)$

(Blei et al., 2003).

θ is the topic distribution of the document. It is a vector where the i-th entry represents to what ratio the document consists of topic i. The sum of all entries is exactly 1. α is a hyperparameter of the Dirichlet distribution ($Dir(\alpha)$) from which the topic distribution is chosen. It modulates how sparse θ the topic vector is or how unequal the shares of different topics are distributed. β is a matrix of shape $k \times V$ with k being the number of topics and V the size number of unique words in all documents. The word w_n is picked from the distribution given by the row from β corresponding to the topic z_n.

This assumption how documents are generated obviously is not accurate for human generated texts. But under the assumption that the documents, on which topic modeling should be done, were generated like this it is possible to infer θ for each document and β for all words. We only need to set k the number of topics in advance. α can be set as well but usually is set to $\frac{1}{k}$. Then, we get for each document a topic distribution and for each topic a distribution over the words.

This word distribution can work as a way to represent each topic by using the most probable words for each topic as the representation for the topic. Documents are not assigned to one topic but rather described as a combination of multiple topics.

Lastly, it needs to be kept in mind that the generation model works under the assumption that the words are chosen independently of each other, without any sense of context.

2.1.5 BERTopic

BERTopic is a topic modeling method proposed by Grootendorst, 2022. First, all the documents are embedded.

Document embedding is the process of assigning a vector representation to a document. This is done utilizing sentence embeddings. Most state of the art sentence embedding methods are based on transformers (Vaswani et al., 2017). Transformers are a deep learning architecture utilizing an attention mechanism to find embeddings for tokens (tokens are single words or subwords) that are context-dependent which is an advantage compared to static word embeddings. From this basis different techniques exist to create embeddings for whole texts (sentences) based on the embeddings of the token. While simple methods such as averaging over all token embeddings to create a sentence embedding exist, more sophisticated apporoaches like the Sentence-BERT-Framework (Reimers and Gurevych, 2019) usually yield better embeddings. This is because

Sentence-BERT employs an additional pooling layer to create the vector for the entire input text. All sentence embedding models can be used for BERTopic. Grootendorst, 2022 originally proposed using the Sentence-BERT Framework(Reimers and Gurevych, 2019) to embedd the documents.

Then, dimensionality reduction is done. Dimensionality reduction is the task of transforming high dimensional data into a vector space with fewer dimensions. This technique has several advantages including reducing the size of the data, filtering out noise and escaping the curse of dimensionality. The curse of dimensionality describes the problem that relative distances in high dimensional spaces becomes less expressive. The UMAP dimensionality reduction algorithm (McInnes et al., 2018) is used by default to reduce the dimension of the document embeddings down to two. Other dimensionality algorithms such as PCA or TSNE (Van der Maaten and Hinton, 2008) can be used as well.

After this the reduced embeddings get clustered. Clustering is the task grouping similar datapoints together. The similarity measure used is usally a distance metric or cosine similarity. The groups into which the datapoints are divided are called clusters. Some algorithms require the number of clusters as parameters while others find a feasible number of clusters themself. Clustering usually is done unsupervised (without ground truth). The reduced embeddings get clustered using the HDBSCAN algorithmMcInnes et al., 2017. For this clustering algorithm the number of clusters is not given as parameter but rather found by the algorithm that only takes a minimum size for the clusters as parameter. Also some data points (in our case document embeddings) are declared to be noise or outlier and not assigned to any cluster by this clustering approach. Also for each datapoint a probability that it belongs to the assigned cluster can be obtained (outliers have probability zero).

The clusters of document embeddings found by the HDBSCAN algorithm are considered to be Topics. This means that some documents are considered outliers and the number of Topics was found not set a priori.

To find representations for the Topics a modified version of the TF-IDF score (Joachims et al., 1997) is used.

$$W_{t,c} = tf_{t,c} \cdot \log\left(1 + \frac{A}{tf_t}\right) \tag{2.1}$$

To compute the importance $W_{t,c}$ of term t for cluster (topic) c all documents assigned to the same cluster are concatenated together. The term frequency $tf_{t,c}$ is defined as the frequency of term t appearing in the concatenated documents in cluster c. A is the average number of words per cluster. tf_t is the frequency of term t across all clusters. With measure the terms in a cluster can be ordered by importance, and the top n terms then are used to represent the Topic.

The number of topics found by the clustering cannot be set by the user, but the user can set a number of topics they expect from the BERTopic run. If this number is higher or equal than the number of clusters found by HDBSCAN, nothing is changed, still only the clusters by HDBSCAN are returned as topics. But if the user expects fewer topics than found clusters, iteratively the smallest cluster is merged with the cluster having the most similar vector of c-TF-IDF scores until the number of topics specified by the user is reached.

Besides representing the topics by the most important terms from them, the topics can be represented by the documents with the highest probability of belonging to the topic given

by HDBSCAN, or by the highest cosine similarity to the mean of all document embeddings contained in the Topic.

In contrast to LDA BERTopic does not have to use the bag-of-words assumption to find the topics, but it should be noted that the representation of the topics still relies on the bag-of-words assumption. For the merging of topics this assumption is needed as well.

2.1.6 Topic GPT

TopicGPT is a fairly new topic modeling method developed by Pham et al., 2024. TopicGPT uses Large Language Models (LLMs) to generate fitting topics and assign documents to one of the topics. Large Language Models are models that generate new text based on an input text. LLMs use transformers Vaswani et al., 2017 to contextually embed the tokens of the input text and then they utilize a neural network to predict the next token. This next token then can be added to the input and the following token can be predicted in the same way. Through this method iterativly an output text is generated. The initial input of a LLM is called prompt and the output text is often called answer.

Assigning a document to a topic picked from a list of topics is straightforward. The prompt just contains the topic list and the document. This is just an unsupervised text classification task.

But before this simple step of the topic modeling can be done, the list of Topics needs to be generated. To do this, the user gives a small number of example topics (two are according to Pham et al., 2024 already sufficient). Then, the LLM is prompted iteratively with a prompt containing the current list of topics (in the beginning the example Topics from the user) and one document from the corpus. The instruction in the prompt is to either assign the document to an existing topic from the list or to add a topic to the topic list into which the document fits well. This is done with either a subset of documents or with all documents if the required resources are available to the user.

After this step there exists a list of topics, but to ensure that the list is not redundant the topic list gets refined. To do this all topic labels are embedded using Sentence-BERT (Reimers and Gurevych, 2019). Then, pairs of topic label embeddings with small cosine similarities are found. Small groups of those pairs of similar topic labels are then again given to a LLM with the instruction to merge similar topic labels, if they are too redundant.

After this refinement the list of topics is finalized. Then, the LLM is prompted with one document at a time and the list of topics to assign the document to exactly one topic from the list.

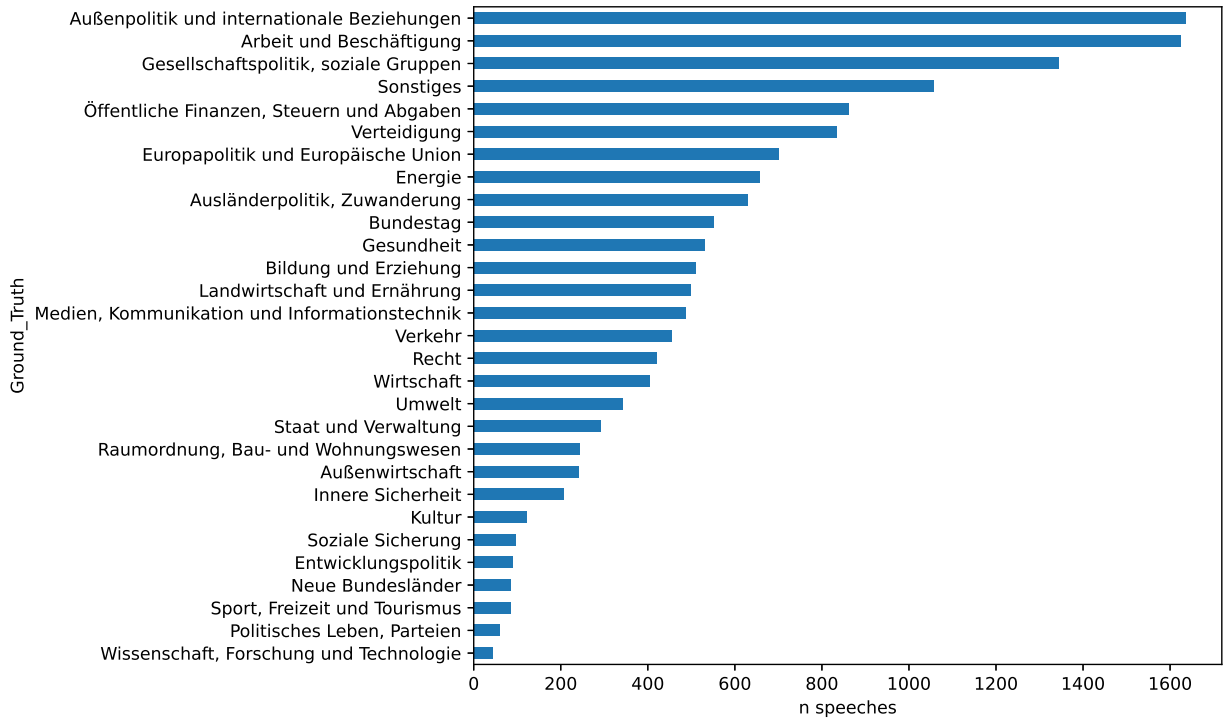

Figure 2.1: The number of speeches assigned to each of the 29 Topics by the human labelers

2.2 Experiments

2.2.1 Dataset Offenes Parlament

Offenes Parlament Dataset	
number of topics	29
number of speeches	15109
median words per speech	724
mean words per speech (stdv)	772.52 (380.90)
timespan of speeches	10/2013 - 10/2017

Table 2.1: Description of the Offenes Parlament dataset

For evaluation of the topic modeling, I used a dataset provided by Offenes Parlament[1]. The dataset contains all 15109 speeches given in the Deutsche Bundestag during the 18th electoral Period (2013 - 2017). The dataset also contains the agenda item each speech was held on. Offenes Parlament then manually assigned each agenda item to a set of broader topics. The List of possible topics was created by Offenes Parlament and based on a list from the Bundestag itself. In total, the List contains 29 topics. Table 2.2.1 shows that on average each speech contains 773 words. The high standard derivation shows that the length of speeches varies a lot between the speeches.

1. https://offenesparlament.de/ accessed March 9, 2025

Because for my use case I want to assign each speech to exactly one topic but Offenes Parlament assigned each agenda item multiple topics, I declared the first mentioned topic as the most important one and therefore it is considered as the ground truth gold label of that speech. So through joining together the speeches with the list of agenda items and their assigned topics, I obtained a dataset of Bundestag speeches assigned to topics which I will use for evaluation of the different methods. Figure 2.1 shows how many speeches are assigned to each of the 29 topics in the final dataset described here.

2.2.2 Metrics

To evaluate the results from the different topic modeling methods, I used multiple evaluation metrics.

V-measure (Rosenberg and Hirschberg, 2007) compares the inferred labels of the clustering to gold class labels. It calculates the homogeneity of the assignment as:

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases} \tag{2.2}$$

with $H(C)$ being the entropy of the gold classes and $H(C|K)$ beiing the calculated as:

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}} \tag{2.3}$$

with a_{ck} being the number of datapoints of class c assigned to cluster k. Similar to this completeness of the assignment is calculated as:

$$c = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases} \tag{2.4}$$

with $H(K)$ being the entropy of the clusters and $H(K|C)$ being calculate as:

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}} \tag{2.5}$$

The v measure score combines both homogeneity and completeness to asses the clustering quality:

$$V_\beta = \frac{(1+\beta)\cdot h\cdot c}{\beta\cdot h + c} \tag{2.6}$$

where β is a parameter that weights homogenity vs completenes. By default β is 1 which gives equal weight to both.

Mutual Information is calculated as:

$$MI(C,K) = \sum_{i=1}^{|C|} \sum_{j=1}^{|K|} \frac{\left|C_i \cap K_j\right|}{N} \log \frac{N\left|C_i \cap K_j\right|}{|C_i| \cdot |K_j|} \tag{2.7}$$

With $|C|$ and $|K|$ being the number of inferred clusters (C) and gold label classes (K). C_i being all datapoints assigned to cluster i and K_j all datapoints with gold label j. This score measures the similarity between the cluster and the gold labels, it can also be used if there are no gold labels available to compare two different clusterings.

Both V-measure and mutual info are useful because they do not require a mapping from the inferred clusters to the gold classes. This is helpful because LDA and BERTopic describe the topic with a list of keywords found in the documents and not with one of the 29 topic names assigned by Offenes Parlament to the documents not as one concise phrase, as the Ground Truth Offenes Parlament data does.

For that reason, accuracy, precision and recall for BERTopic and LDA cannot be immediately calculated with the dataset from Offenes Parlament because BERTopic and LDA represent the topics as lists of keywords. Therefore, I decided to use LLMs to map the list of keywords found by LDA and BERTopic to the list of topics from the Ground Truth. In the prompt, I gave the LLMs the list of keywords for a topic and two documents assigned to that topic and the list of ground truth topic names to pick from (the whole prompt can be found here Appendix Listing A.3). I call this method *topic mapper* because it maps the Offenes Parlament topics onto the topics from BERTopic and LDA. To assure that this approach was valid, I did the mapping manually for 20 topics found by BERTopic and 20 topics found by LDA and compared it to 3 Large Language Models from different companies doing the mapping with the same prompt. I used the following LLMs:

- **LLama-3.1-8B-Instruct** [2] was developed by Grattafiori et al., 2024. The model will be referred to as *Llama3.1* in this thesis. *LLama3.1* has a context lenght of 128k tokens and has 8 billion parameters. It was trained on a multilingual corpus of about 15 trillion tokens. Llama3.1 is used because of it is open source and because it performs well for its size in established metrics. Also its comperably long context length is useful.
- **Mistral-7B-Instruct-v0.3** [3] was developed by Jiang et al., 2023. This model will be called *Mistral7b* in this thesis. *Mistral7b* has 7 billion parameters and a context length of 8192 tokens. The model was chosen because it is open source and well established.
- **gemma-2-9b-it** [4] was created by engineers at google. In this thesis it willy be called *Gemma2*. *Gemma2* has 9 billion parameters a context length of 4096 tokens and was trained on a primarily english dataset containing about 8 trillion tokens. The model was chosen because it is open source and resource efficient.

I used instruct models because they were fine-tuned to follow instructions and the prompts I will give them contain clear instructions. As seen in Table 2.2 for the mapping of BERTopic's representations *Gemma2* had the best accuracy and for LDA's representations Llama3.1 performed best. For this reasons the *topic mapper* uses *Gemma2* to map the Offenes Parlament topics onto the BERTopic topics and *Llama3.1* is used for LDA.

2. https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct accessed March 10, 2025
3. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3 accessed March 10,2025
4. https://huggingface.co/google/gemma-2-9b-it accessed March 10,2025

Accuracy accuracy is the percentage of correctly inferred labels. I calculate two accuracy values. First, the assignment is counted as correct if it is identical to the gold label. The gold label is the first topic listed by Offenes Parlament. But Offenes Parlament provides multiple topics for each speech. Therefore I secondly calculate the percentage of topic assignments, where the assigned topics is among the Offenes Parlament labels. This accuracy will be called accuracy over all in the results section.

precision, recall, f1 are standard classification metrics. Fundamentally those metrics are defined only for binary classifications into positive and negative. They are calculated based on a confusion matrix that describes how many datapoints were classified correctly as positives (True Positives), correctly as negatives (True Negatives), falsely as negatives (False Negatives) and falsely as positive (False Positive). precision and recall are calculated as precision $= \frac{\text{TP}}{\text{TP+FP}}$ and recall $= \frac{TP}{TP+FN}$. With TP being the number of true positives, FP being the number of false positives and FN the number of false negatives. The f1 score is the harmonic mean of precision and recall. There are different methods to apply those metrics to multi class classifications like my use-case requires. I choose calculating the scores for each class independently (The selected class is the positive all other classes are the negatives) and then computed an average weighted by the size of the classes from the independently calculated scores.

Silhouette coefficient (Rousseeuw, 1987) To evaluate the quality of clustering done by BERTopic I used the Silhouette Coefficient s. The Silhouette can be used without a Ground Truth Label but needs a position not only an assigned label, which is the reason I can only use it on the BERTopic results. The Silhouette coefficient takes into account the mean intra-cluster distance of each datapoint and the mean nearest cluster distance. The complete formula for the silhouette coefficient s can be seen here:

$$s = \frac{\sum_{i}^{n_{samples}} \frac{(b_i - a_i)}{max(a_i, b_i)}}{n_{samples}}$$

with a_i being the mean distance between datapoint i and all points that are assigned to the same cluster as datapoint i and b_i being the mean distance between datapoint i and all datapoints that are assigned to the next nearest cluster that i was assigned to. After initial testing, I found that the silhouette coefficient, when computed with the complete embeddings of the speeches, was not differentiating between the different versions of BERTopic I tested. For that reason I computed the Silhouette Coefficient on the vectors obtained after the dimensionality reduction which is part of the BERTopic method.

Model	Acc	Error rate	Model	Acc	Error rate
Llama3.1 8b-it	55%	**7.14%**	Llama3.1 8b-it	**45%**	**7.14%**
Mistral 7b -instruct	60%	**7.14%**	Mistral 7b -instruct	35%	10.71%
Gemma2-9b-it	**65%**	**7.14%**	Gemma2-9b-it	30%	17.86%

Table 2.2: The representation mapping accuracy (Acc) of different LLM models compared to manual labeling. As error counts either an error while parsing the answer from the LLM or if the answer is not part of the Ground Truth topic list. Left with the representations from BERTopic right with LDA

2.2.3 Experiment Setup

I split the dataset into development and test data to avoid overfitting while fine-tuning different model parameters.

LDA For LDA there were not many parameters to modify. I used an implementation from scikit-learn [5]. Through testing with the development dataset I decided to remove all words, which appear in more than 20% of all documents, plus general German stopwords obtained from the Natural Language Toolkit (NLTK)[6]. I set the number of topics to 29, because it is the number of real gold topics in the evaluation dataset.

BERTopic For BERTopic I used the implementation by Grootendorst, 2022. I fixed the number of topics to 29+1, one more than in the evaluation dataset because BERTopic always creates one outlier topic. The speeches assigned to this outlier topic can be assigned to another topic. To do this each outlier speech gets embedded again and this embedding get compared via cosine similarity to the mean embedding of the non outlier topics. The outlier speech then is assigned to the topic to which mean embedding it has the highest cosine similarity. While fine-tuning some parameters of the clustering algorithm and BERTopic itself, the main focus was to test how different embedding models impact the overall performance. The standard implementation of BERTopic uses the Sentence-BERT-Framework (Reimers and Gurevych, 2019). The Massive Text Embedding Benchmark (MTEB) Leaderboard (Muennighoff et al., 2023) gives a compilation of the performances of different embedding models, broken down by task. Topic modeling, when performed by BERTopic, can be seen as a clustering task, therefore I picked models that performed well in the german clustering part of the MTEB (Wehrli et al., 2023). I picked the following embedding models to test with BERTopic:

- **paraphrase-multilingual-MiniLM-L12-v2**[7] (Reimers and Gurevych, 2019) This modell will be called *paraphrase* for this thesis. It has 118 million parameters was trained on a multilingual corpus. Its context length is only 128 tokens long. The model was chosen as baseline because it is the default model used by BERTopic.
- **jina-embeddings-v3**[8] (Sturua et al., 2024) this model will be called *jina* during this thesis. *jina* has 570 million parameters was trained on multilingual data and supports a context length of up to 8192 tokens. The model was chosen because of its good performance in the MTEB leaderboard for german clustering and because it is open source.
- **sentence-t5-xxl** (Ni et al., 2022)[9] This model is an extension of the T5 model(Raffel et al., 2020) with 11 billion parameters and no token limit. It will be reffered to as *sentence-t5* during this thesis. It was trained on english data. Although it was trained on english data it performed well in the MTEB leaderboard for german clustering, which is why it was chosen.

5. https://scikit-learn.org/ accessed March 10, 2025
6. https://www.nltk.org/ accessed March 9, 2025
7. https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 accessed March 10,2025
8. https://huggingface.co/jinaai/jina-embeddings-v3 accessed March 10, 2025
9. https://huggingface.co/sentence-transformers/sentence-t5-xxl accessed March 10, 2025

Model	V-measure	Mutual info
LDA	**23.08**%	**0.7155**
BERTopic(paraphrase)	32.58%	1.0092
BERTopic(sentence-t5-xxl)	35.92%	1.1167
BERTopic(German_Semantic_STS_V2)	40.63%	1.2623
BERTopic(jina-embeddings-v3)	**40.99**%	**1.2704**
Own Prompt(Gemma2-9b-it)	34.83%	1.1119
Own Prompt(Llama3.1-8b-it)	32.07%	1.0744
Own Prompt(Mistral7b-it)	**36.44**%	**1.2169**
TopicGPT (Mistral/Gemma)	**39.15**%	**1.2081**

Table 2.3: The results of the different approaches for the metrics that just compare the assignments independent of the labels (v-measure and mutual info score)

- **German Semantic STS V2**[10] The model will be called *German-STS* in this thesis. This model is based on gbert-large (Chan et al., 2020) but was refined for semantic similarity tasks. It has 335 million parameters and a maximum number of tokens of 512. It was chosen because it perfomed well in the german MTEB leaderboard for clustering and because it was trained on german data.

BERTopic is typically used for shorter documents like social media posts (Hellwig et al., 2024). The author Grootendorst, 2022 also mentions that it performs best for short documents. Hence I divided the speeches from the Offenes Parlament dataset into paragraphs and let BERTopic consider each paragraph as a separate document. Then, I assigned a whole speech to the Topic that most of its paragraphs were assigned to.

TopicGPT For TopicGPT, I used the prompts and implementation by Pham et al., 2024 with minor adjustments. But while TopicGPT, as the name suggests, usually is used with GPT4.0 (OpenAI et al., 2024) I used *Mistral7b*, because it is as open-source model available. TopicGPT finds topics that describe the corpus well. To calculate the classification metrics those extracted topics need to be mapped onto the Offenes Parlament topics. This is done using the same *topic mapper* that was used to map the Offenes Parlament topics onto the topics found by LDA and BERTopic.

TopicGPT's main idea is to use a prompt to generate the Topics the documents should be assigned to. Given the fact that through the Ground Truth dataset 29 Topics are already known, I tried the direct way of just asking different LLMs to assign each speech to one of the Ground Truth Topics. The prompt to do this can be found in Appendix Listing A.1. I will refer to this approach as Own Prompt from now on

To do this I used *Gemma2*, *Mistal7b* and *LLama3.1*

Model	Acc	Acc (all)	Prec	Recall	f1
LDA	**17.02%**	**25.80%**	**14.42%**	**17.02%**	**14.69%**
BERTopic(paraphrase)	25.81%	38.75%	31.42%	25.81%	25.53%
BERTopic(sentence-t5-xxl)	30.65%	47.86%	30.08%	30.65%	27.60%
BERTopic(German_Semantic_STS_V2)	27.89%	45.19%	27.79%	27.89%	25.78%
BERTopic(jina-embeddings-v3)	**33.43%**	**51.39%**	**28.25%**	**33.43%**	**28.34%**
Own Prompt(Gemma2-9b-it)	**34.75%**	**54.21%**	**46.66%**	**34.75%**	**36.41%**
Own Prompt(Llama3.1-8b-it)	31.73%	49.13%	40.87%	31.73%	33.39%
Own Prompt(Mistral7b-it)	34.33%	53.80%	44.23%	34.33%	35.03%
TopicGPT (Mistral/Gemma)	31.50%	44.17%	45.16%	31.50%	32.89%

Table 2.4: The classification metrics of the different approaches compared. Accuracy (Acc), precision (Prec), Recall and f1 score are calculated based on the gold label, accuracy over all (acc(all)) is over all topics Offenes Parlament assigned.

Embedding model	Silhouette score
paraphrase multilingual	0.1723
sentence-t5-xxl	0.1947
German Semantic STS V2	0.3410
jina-embeddings-v3	**0.3877**

Table 2.5: The silhouette scores for the BERTopic reduced embeddings when used with different embedding models paraphrase multilingual is the default model used for BERTopic (Reimers and Gurevych, 2019)

2.2.4 Results

Table 2.3 shows the results for the clustering metrics v-measure and mutual information. Table 2.4 shows the classification metrics accuracy, precision, recall and f1 score for all approaches. As seen in those results, LDA is outperformed significantly by BERTopic and the LLMs with the own Prompt in every metric. This shows how modern approaches to natural language processing not relying on the Bag of Words assumption but taking into account context are superior to the approach of assuming conditional independence between all words in a text.

All BERTopic variants outperform LDA in every evaluated metric. Among the different embedding models *jina* (Sturua et al., 2024) performed best in all metrics including the silhouette score shown in Table 2.5 that was only evaluated for BERTopic. The smallest model *parahrase* (Reimers and Gurevych, 2019) performed worst but the biggest tested Model *sentence-t5* scored only marginally better and was outperformed by the two smaller models *German-STS* and *jina*.

Among the LLMs *Mistral7b* had the lowest error rate as shown in Table 2.6. The errors counted as wrongly classified. *Mistral7b* also scored best for v-measure and mutual info among the LLMs. For the classification scores *Gemma2* scores best.

10. https://huggingface.co/aari1995/German_Semantic_STS_V2 accessed March 9, 2025

Model	Error rate
Own Prompt(Gemma2-9b-it)	16.59%
Own Prompt (Llama3.1-8b-it)	13.71%
Own Prompt (Mistral7b-it)	**9.49%**
TopicGPT (Mistral7b/Gemma2)	**17.17%**

Table 2.6: The error rates of the LLM methods. Both when using only a prompt to classify and when using TopicGPT

TopicGPT seems to work similar well than the LLMs with own Prompt and BERTopic, but in the topic generation phase it generated half as many Topics as there were documents in the corpus. After that, *Gemma2* maps those topics onto the 29 ground truth topics. Because that mapping receives also two sample documents, *Gemma2* essentially does the classification instead of the original TopicGPT system. Pham et al., 2024 mentions that the topic generation works bad with smaller language models than GPT4.0 OpenAI et al., 2024 , which has 1.8 trillion parameters.

> Mistral and GPT-3.5-turbo produced 1,418 and 151 topics, respectively. [...]. Additionally, most of the generated topics are overly specific with a low frequency of occurrence

(Pham et al., 2024)
Given the large number of topics generated by TopicGPT in my testing with *Mistral7b*, I can confirm that.

Overall for the classification metrics *Gemma2* scored the best, which might be because it is the largest tested LLM and the LLMs with the own prompt essentially are a classifying system, so it makes sense that they perform well for the classification metrics. The best clustering metrics v-measure and mutual info were reached by BERTopic with *jina*, This is plausible because BERTopic works with the whole corpus to find clusters, while the LLMs only see one document at a time. Therefore the overall quality of the clustering is better than the LLMs that cannot pay attention the cluster coherence.

2.2.5 Discussion

While doing the manual labeling of the BERTopic and LDA representations to evaluate the accuracy of the LLM representation mapper (Table 2.2), I subjectively found the BERTopic representations of the topics more descriptive than the representation extracted by LDA, which also indicates the superiority of BERTopic over LDA. An example of this can be seen in Figure 2.2 where two representations for the topic *Landwirtschaft und Ernährung* (Argiculture and nutrition) are shown. Both have overlapping keywords but LDA has some words that do not directly relate to the topic like *minister (secretary)*, *entwicklung (development)*, *unternehmen (company)* while the keywords from BERTopic all align well with the topic.

While *sentence-t5* was the model with the most parameters among the tested sentence-embedding models its performance was mediocre. This can be explained by the fact that it is trained on english text while all other models are trained on multilingual text and in the case of *German-STS* even fine tuned for german text in particular.

LDA representation

minister verbraucher ländlichen verbraucherinnen entwicklung verbraucherschutz unternehmen bauern tierschutz tiere

BERTopic representation

ernährung verbraucher tierschutz tiere bauern lebensmittel gentechnik landwirte pflanzen gentechnisch

Figure 2.2: Comparison of the representations found by LDA and BERTopic. I manually assigned both to the label *Landwirtschaft und Ernährung (Argiculture and nutrition)* but the BERTopic representation makes this classification easier.

The bad performance of TopicGPT, caused by the usage of mid-tier LLMs instead of the flagship GPT4.0 (OpenAI et al., 2024) model, expressed itself in the huge number of topics generated in the topic generation step. This problem could be addressed if instead of one topic refinement step, where similar topics can be merged, more topic refinement steps would be performed iteratively. While this might help bring down the number of generated topics, it also would increase the resource usage because of the multiple refinement runs. If this approach would be tested, the overall resources used should be compared to the resources needed if a flagship LLM with only one refinement iteration was used.

Although the LLMs with the own prompt performed well in the classification metrics, it needs to be kept in mind, that the the task for the LLMs with the own prompt was easier because they got the list of ground truth topics during their assignment process, while all other methods just got their extracted topics mapped to the ground truth. The goal of the topic Modeling for my web application is to find coherent topics and not to assign speeches to Topics some expert defined a priori. For this reason the LLMs with the own prompt will not be used in my final application, despite their good performance.

BERTopic scored the best in the clustering metrics. This shows that the clusters of speeches identified by BERTopic are aligning well with human assigned topics. BERTopic extracts the topics without the need for a priori knowledge of them, and as Figure 2.2 showed the representation of topics is more expressive and coherent than LDAs representations. That is why I will use BERTopic to improve the existing web application. But although the representations of BERTopic are better than LDA the format of a list of keywords as representation, is not immediately understandable for a unbeknownst user. For this reason I will use a LLM to assign a name to each topic extracted by BERTopic, similarly to the *topic mapper* used to assign Offenes Parlament labels during this section.

3 Stance Detection

3.1 Introduction

3.1.1 Motivation

In political contexts, the position of political actors is one of the most important variables. The position of political actors on topics allows prediction about the actions of this political actor. In democracies people cast their vote based on the positions of the actors on different topics. For this reason extracting the positions of political actors is important. Finding the position of the public, expressed for example on social media is relevant as well, because politicians orient themself on the public opinion. Currently one task of journalism is to carve out the position of political actors, and pollsters create polls to find the public opinion. This shows that there is a big interest in both the political positions and the public opinion. The goal of stance detection is to extract those positions from text bases, making the tasks of finding the positions more accessible and faster.

For my web application, stance detection can help to create an understanding of how the different parliamentary actors view a topic. A user who is interested in a certain topic should be able to find the stance of different parliamentary actors on it. This allows the user to gain information about the stances fast and helps them to understand the proceedings in the Bundestag.

3.1.2 Task Definition

Mohammad et al., 2017 defines stance detection as follows: „Stance detection is the task of automatically determining from text whether the author of the text is in favor of, against, or neutral toward a proposition or target" So stance detection always estimates the stance towards a target. This target needs to be defined a priori. This definition narrows down to the stances of favor, neutral or against. But stance detection can also be defined broader as finding the stance of a text towards a target. With the stance being either a discrete variable (like in favor,neutral, against) or a continuous scale between two extremes (like pro-contra or left-right)

This brings stance detection closer to the adjacent task of political scaling.

3.1.3 Goal

The goal is to find an approach to stance detection that allows us to extract a meaningful continuous scale of the stances by the political actors in the Bundestag. To do this I will use political scaling methods. Political scaling tries finding an axis onto which the documents from a corpus can be scaled. This axis then should resemble a meaningful policy dimension like the left-right scale, or favor-against. This allows us to utilize political scaling for our goal of finding stances on the topics, extracted with BERTopic.

3.1.4 Wordfish

Wordfish is a scaling approach specifically developed for political positions by Slapin and Proksch, 2008. Similarly to LDA, Wordfish works under the Bag of Words assumption, so it only looks for the number of times a word appeared in a document, not where or in what context it appeared. Wordfish assumes that the number of times y a word j occurs in a document i is drawn from this Poisson Distribution:

$$y_{ij} \sim Poisson(\lambda_{ij})$$
$$\lambda_{ij} = exp(\alpha_i + \psi_j + \beta_j + \omega_i)$$

Where α_i and ψ_j are document and word fixed effects. The fixed effects are there to compensate for different document lengths and higher frequency of certain words in all documents. The scaling results are: ω_i the estimated position of document i and β_j is an estimate of the word's significance in discriminating between the two ends of the scale. To estimate those latent variables, an expectation maximization algorithm (McLachlan and Krishnan, 2007) is used until the latent variables converge. Then, the ω_i variable is expected to represent a meaningful scale for the positions. This expectation hinges on the assumption that the frequency with which a document uses different words is an indication of its political position. Besides the Naive Bayes assumption that words are used independently of each other, Wordfish also assumes that the meaning of a word does not change between documents. This is especially important to keep in mind when scaling documents from different points in time together, as such changes become more likely the more time is in between the publication of the documents that are scaled together. Wordfish also returns a standard derivation obtained from parametric bootstrapping.

3.1.5 Glavaš method

Glavaš et al., 2017 proposes a method to spread a corpus of political texts across a one dimensional scale. This is similar to the goal of the Wordfish algorithm. Glavaš method , contrary to Wordfish, is created to handle multilingual corpora but works also for monolingual corpora. The basic idea of Glavaš method is that it uses Embeddings to give each pair of documents from the corpus a similarity score. This similarity score is then considered to be the weight of an edge connecting the two documents the similarity score is computed for. This creates a complete undirected weighted graph were the documents are the vertices and the similarity scores the weight of its edges.

Then, it is assumed „that the two semantically most dissimilar texts, which we name pivot texts, represent the opposite position extremes for the political dimension of interest. We initially assign them extreme position scores of -1 and 1." Glavaš et al., 2017. With this assumption and the Graph created from the similarity measure, the semi-supervised method Harmonic Function Label Propagation (HFLP) can be used to find position estimates of each document. Label propagation methods work on graphs where the label of some but not all vertices is known. Using the assumption that a stronger connection between vertices results in a higher probability that those vertices share the same label, label propagation algorithms infer the labels of the unknown vertices. HFLP works with the assumption that the label of a vertex should be the weighted average of the labels of its adjacent vertices. In our case the average gets weighted by the similarity between the two vertices. The labels inferred by HFLP are then interpreted as positions of the non-pivot texts.

In the end, the Glavaš method rescales the position $p(t)$ of pivot text t in the following way:

$$p(t) = \sum_{t_i \in NP} p(t_i) \cdot s(t, t_i) \tag{3.1}$$

where NP is the set of the non-pivot texts, $p(t_i)$ the estimated position of text t_i and $s(t, t_i)$ the similarity between t and t_i (Glavaš et al., 2017). This is done because „(1) our metrics of semantic similarity are imperfect, i.e., the scores they produce are not the gold standard semantic similarities, but even if they were (2) we do not know to what extent the semantic similarity we measure correlates with the particular political dimension being analyzed" (Glavaš et al., 2017)

The two similarity measures Glavaš method proposes both use word embeddings as basis. **Aggregation similarity:** This measure simply finds an embedding for the whole document by averaging over the L2-normalized word embeddings of all the words in the document. The similarity between two documents then is defined as the cosine similarity of the two document embeddings created earlier.

Alignment similarity: This measure works by pairing together the most similar words, judged by the cosine similarity of their embeddings. Once a word is paired up, it cannot be paired with another word. The similarity between two documents then is calculated as the mean of all the cosine similarities of the pairs of words. Glavaš et al., 2017 notes that this measure is similar to the METEOR score (Lavie and Denkowski, 2009). This similarity measure also resembles the later by Zhang* et al., 2020 presented bert-score.

The final step to obtain the similarity scores in both cases is to rescale them to be between 0 and 1.

3.1.6 Embscal

Embscal is an approach to political scaling that, similarly to BERTopic, uses documents embeddings and dimensionality reduction to obtain a meaningful scale to order the documents on. All documents are embedded into a high dimensional vector space. The resulting embeddings encode the information from the texts. Dimensionality reduction tries to project high dimensional data to a low dimensional space that captures the most important information from the high dimensional space. Assuming the policy axis we want to scale the texts onto, is the strongest differentiator between the documents, dimensionality reduction down to a one dimensional vector space is done. This one dimensional scale then is the resulting policy axis. It remains to be tested if that assumption holds for our dataset, and if the resulting scale correlates with a relevant political dimension like the ideological left-right dimension.

3.2 Experiments

To evaluate Wordfish, Glavaš method and Embscal, I tested them on German parties election manifestos. All of the methods estimate the position of documents on a one dimensional scale, and because party manifestos are widely regarded as fundamental statement about a parties overall ideological values I expect this scale to strongly correlate with the ideological left-right scale.

3.2.1 Dataset

Manifesto Project dataset	
number of manifestos	33
median words per manifesto	26786
mean words per manifesto (stdv)	28727.27 (17059.84)
timespan of manifestos	1998 - 2017
CHES Leftmost position	1.23 (DIE LINKE -2013)
CHES rightmostposition	9.24 (AFD-2017)

Table 3.1: Description of the Manifesto Project dataset

I created my evaluation dataset by combining two sources. I used the German parties manifestos as collected by the manifesto project database[1]. I used their API to fetch the plain text of the parties manifestos and joined that with some metadata also provided by the manifesto project like publishing date and party name. I then used the Chapel Hill expert survey trend file (Jolly et al., 2022) to get the ideological left right position of the parties. The Chapel Hill Expert Survey „contains measures of national party positioning on European integration, ideology, and several European Union (EU) and non-EU policies for six waves of the survey, from 1999 to 2019." (Jolly et al., 2022). Those measures are generated by averaging over the estimation of the position by multiple political experts. I used the left-right estimate from the survey which gives a measure between 0 (Extreme Left) and 10 (Extreme Right) for each party and each year. I matched each estimate with the parties manifesto that was published the shortest time before the estimate took place. I consider this combination valid because party manifestos are a synthesis of the parties' positions on multiple topics therefore overall they should resemble a statement about the general parties ideological position on the left right scale. I thereby obtained 34 plain text party manifestos together with a gold position ideological score between 0 and 10 for each manifesto. The 34 manifestos are assigned to 6 different Surveys from 1999, 2002, 2006, 2010, 2014 and 2019 Tabel 3.2.1 shows some statistics about the dataset. The manifestos are quite long but their length has a wide spread.

3.2.2 Experiment Setup

I used a random scaler that assigns each manifesto a random float between 0 and 1 as a baseline. This is done to help judge the performance of the other approaches.

Wordfish To run Wordfish I used the implementation provided by Slapin and Proksch, 2008. I preprocessed the manifestos texts by removing German stopwords and by removing German party names and party abbreviations because I wanted to prevent Wordfish from scaling just based on the parties' names. I tested giving Wordfish the manifestos before and after lemmatizing them with the lemmatizer from the Spacy python package [2].

1. https://manifesto-project.wzb.eu/ accessed on March 9, 2025
2. https://spacy.io/ accesed March 9, 2025

Glavaš method I implemented Glavaš method myself and tested both similarity measures. To ensure comparability to Wordfish, I also removed all parties' names from the documents, but replaced them with the placeholder 'Partei' (party) to keep a valid sentence structure in the documents, Glavaš method uses GloVe word embeddings (Pennington et al., 2014), which are static. Static word embeddings are not able to model context. This is why I decided to use modern sentence embeddings instead. I picked the multilingual-e5-large-instruct sentence embedding model from now on called *mutlilingual-e5* from Wang et al., 2024. The model has has a maximum context length of 512 tokens, 560 million parameters and is based on the xlm-roberta-large model (Conneau et al., 2020). The reason I picked this model is that it is open source and that it was trained on a multilingual corpus. Also the model was among the top scoring models for the semantic textual similarity benchmark on the MTEB-Leaderboard (Muennighoff et al., 2023) in the multilingual section. I also tested *jina* (Sturua et al., 2024) and *parahrase* (Reimers and Gurevych, 2019). Both of them were already tested in the topic modeling chapter. The Manifesto project divided the manifestos already into clauses. I embedded those clauses with the different sentence embedding models instead of embedding single words like Glavaš method did. From there on the Glavaš method is run without further modifications. I tested both the aggregation similarity (AVG) and the alignment similarity (ALIGN).

Embscal I also replaced party names with 'Partei' for comperability to the other approaches. For the Embscal method, I picked the *multilingual-e5* model (Wang et al., 2024) from the MTEB semantic textual similarity Leaderboard (Muennighoff et al., 2023).

LLM As a final method, I let different LLMs scale the manifestos. The previously described methods find some axis along which the documents get scaled. After this axis was found it can be observed how well this axis correlates with some policy axis (i. e. left-right). To write a useful prompt for the LLMs we need to decide a priori along which policy axis we want our documents to be scaled. I want to scale them along the ideological left right axis so I wrote two prompts asking the LLM to scale the manifestos with different left right definitions as I could not find a definition in the CHES Survey Jolly et al., 2022 . I prompted once with a definition provided by ChatGPT (The definition can be found in Appendix Listing A.9) and once citing the Wikipedia definition of Left and Right (The whole prompt including the definition can be found in Appendix Listing A.8) I ran this prompt with three LLM models *Llama3.1*, *Gemma2* and *Mistral7b*. I included one manifesto at a time in the prompt. I put the manifesto in between two copies of the prompt as that resulted in the clearest responses. The Manifestos are relatively long. Because *Gemma2* only has a context length of 4096 tokens and *Mistral7b* of 8192 tokens (only Llama3.1 could fit most manifestos in its context length of 128k tokens) it was necessary to truncate the Manifestos to a length that fits the LLMs context length. For comparability I tested all LLM models with truncating the manifestos truncated to 3000 tokens. But separately ran *Mistral7b* and *Llama3.1* with manifestos truncated closer to their maximum context length (7k and 100k respectively). To test if this ability to fit more of the manifesto into the prompt would improve the scaling performance. I always kept the beginning of the manifestos because I assumed that the general left right position of a party becomes clearer in the introduction of the program.

Scaling documents that were published over a large time span (in this case the manifestos publishing dates span around 20 years), together could negatively impact the scaling result of at least Wordfish. For this reason I ran Wordfish once with all manifestos together (OVERALL),

and once I scaled only the manifestos published for the same election together (YEARLY), to avoid the problem of changing word meanings. For comparability, I did this as well with Glavaš method and Embscal. For the LLMs there was no need to do it twice as they only see one manifesto at a time and are oblivious to the other manifestos

3.2.3 Metrics

Evaluating the different methods of positional scaling is done with different metrics. Wordfish and Glavas scale the documents on a axis without assigning which end of the axis is left and which end is right. Because of that I always evaluate both the estimated scaling as given and inverted and take the better result.

I ran Wordfish, Glavaš method and Embscal twice: once only with manifestos from the same year (YEARLY), once together with all Manifestos (OVERALL). To evaluate the YEARLY way, I had to evaluate each year separately because the axis between different years could be inverted and thereby conflicting with each other. To keep the evaluation consistent, I evaluated the LLMs scaling results also once like this and once as a whole. The evaluation scores obtained separately for each survey time then gets averaged to obtain the final score for the method.

The methods estimate the positions on an axis so correlation metrics are a straightforward way to evaluate the estimations compared to the gold positions.

Pearson I use the Pearson correlation coefficient because it assumes a linear relationship between estimations and gold positions, thereby giving insight how well the absolute distances between the parties are captured. It is calculated as follows:

$$r_P = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.2}$$

with x_i being the i-th estimation and y_i the corresponding gold label. $\bar{x}$ and $\bar{y}$ are the means of the predictions and gold labels.

Spearman Spearmans correlation coefficient is used to judge how well the order of the parties is estimated. It is calculated as follows:

$$r_S = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{3.3}$$

where d_i is the difference of ranks between x_i and y_i.

Kendall Kendalls Tau is similar to Spearman in the way that it only assumes a monotonic relationship between prediction and gold labels. It is calculated as follows:

$$\tau = \frac{K - D}{K + D} \tag{3.4}$$

with K being the number of concordant pairs. Concordant pairs are pairs of (x_i, y_i) and (x_j, y_j) where $i < j$ and either $x_i < x_j$ and $y_i < y_j$ or $x_i > x_j$ and $y_i > y_j$. D is the number of pairs that is not concordant (discordant).

Method	failed	r_P	r_S	τ	PA
Random	0	0.1379	0.0834	0.0721	53.41%
Mistral7b (Wiki 4096)	2	0.1522	0.1269	0.1118	47.74%
Llama3.1 (Wiki 4096)	7	0.6749	0.6467	0.502	66.77%
Gemma2 (Wiki 4096)	4	0.5031	0.4993	0.4091	56.16%
Llama3.1 (LLM 4096)	6	**0.7999**	0.7479	0.5636	66.67%
Llama3.1 (LLM 100k)	4	0.5875	0.5977	0.4832	67.24%
Gemma2 (LLM 4096)	8	0.7720	**0.7858**	**0.6381**	**69.00**%
Mistral7b (LLM 20k)	8	0.6265	0.4978	0.3978	56.33%
Mistral7b (LLM 4096)	**1**	0.4506	0.3009	0.2353	51.01%
Glavas(AVG) (multilingual-e5-large-instruct)	0	0.2333	0.2850	0.2020	59.88%
Glavas(ALIGN) (paraphrase)	0	0.3054	0.3542	0.2142	60.48 %
Glavas (ALIGN) (multilingual-e5-large-instruct)	0	0.3954	0.4326	**0.3071**	**65.12** %
Glavas (ALIGN) (jina-embeddings-v3)	0	**0.4267**	**0.4521**	0.2950	64.51 %
Embscal (multilingual-e5-large-instruct)(umap)	0	**0.3157**	**0.2465**	**0.1632**	**57.95%**
Wordfish (unlemmatized)	0	0.7429	0.7703	0.5579	77.65%
Wordfish (lemmatized)	0	**0.7640**	**0.8133**	**0.6414**	**81.82%**

Table 3.2: The scaling performance of the different models, when all manifestos get scaled and evaluated together (OVERALL). r_P and r_S being Pearson and Spearman correlation coefficient. τ is the Kendall-Tau and PA stands for pairwise accuracy

Pairwise accuracy Additionally I use Pairwise accuracy which is the percentage of pairs of position estimates that are in the correct order. This gives an evaluation of how well the relationship between the parties is estimated, similar to Spearmans correlation and Kendalls Tau but clearer to interpret.

Because the LLMs' responses sometimes cannot get parsed successfully, either because it does not provide an estimation at all or because it gives the estimation in an unclear form, I list the error rate of the parsing, which is important to judge the models' performance. The manifestos where the parsing failed get removed from the evaluation, which should be considered when comparing the LLMs scores to the other methods scores because the LLMs could fail estimating the "hard" manifestos and therefore perform better in the evaluation scores.

3.2.4 Results

The results, if all manifestos from all years are scaled together, can be seen in Table 3.2. The first row in the Table is the random baseline. All LLMs have a high error rate. It is clear that the left-right definition written by another LLM performs better than if the Wikipedia definition is used. *LLama3.1* works better when given a to 3096 tokens truncated version of the manifestos. When more of its 128k tokens context length is used to give in most cases the whole manifesto, it does perform worse, though it resulted in less parsing errors. *Mistral7b* on the other hand scored higher in the metrics when given a longer version of the manifestos but also made more mistakes. *Gemma2* could only be run with a highly truncated version of the manifestos because it has a very short context length of 4096 tokens. It performed best among all tested LLMs in

Spearman correlation coefficient, pairwise accuracy and Kendall Tau, while being second best if ranked by the Pearson correlation coefficient. That being said its high failure rate of over 25% somewhat questions its feasibility. Overall, it can be observed that LLMs with a higher failure rate scored higher evaluation scores than the ones with lower failure rate. This could indicate that the LLMs fail to estimate the position of the manifestos, which are the hardest to estimate, thereby boosting their score compared to the LLMs that scale those hard manifestos but do it poorly. Appendix Figure A.2 and Appendix Figure A.3 show how *Gemma2* and *Llama3.1* with the LLM definition and the shorter manifesto version scaled all manifestos.

The Glavaš method performs better than random baseline. Using the alignment similarity (ALIGN) worked better than using the aggregation similarity (AVG). The larger models e5 and jina outperformed the smaller parahrase model. jina performed better for Pearson and Spearman while e5 scored higher for Kendalls Tau and pairwise accuracy. But both embedding models are close together in the scores. Appendix Figure A.5 shows how the Glavaš method with alignment similarity and the e5 embeddings estimates the positions.

The Embscal method performs somewhat better than the random baselin. But the correlation coefficients are still very low and do not indicate a solid ability to estimate the ideological left-right dimension. Among the different tested dimensionality reduction methods, umap performs the best. Appendix Figure A.4 shows how Embscal with umap scaled the manifestos. It can be observed that Embscal scales the manifestos way more continuous than the gold positions.

Wordfish has the highest Spearman correlation coefficient, Kendall Tau and pairwise accuracy among the tested methods and performs well in Pearson correlation coefficient. Because Wordfish does not fail to estimate some positions unlike the LLMs, it can be said that Wordfish correlates the closest with the gold positions from the CHES survey (Jolly et al., 2022). Wordfish performed slightly better when it was run on lemmatized words. The estimations of Wordfish can be seen in Appendix Figure A.1. Like Embscal, Wordfish scales more continuous than the gold positions are. Also, it is noticeable that both manifestos of the AFD Party are estimated to be way more on the left than the gold positions put them. The words that Wordfish identified as most indicative of whether a manifesto is right-wing or left-wing can be seen in Appendix Table A.2.

The results, if only the manifestos from the same year are scaled together, are shown in Appendix Table A.3 because they are similar to the results if all Manifestos are scaled together shown here.

3.2.5 Discussion

The only two methods that scored high enough to be considered usable are the LLM estimators and Wordfish. So only those two should be considered for further application. The big difference between the LLM estimation and the Wordfish estimation is that to write a working prompt for the LLMs the axis, along which to scale, needs to be described with words before the estimation takes places. Ideally you need to give the LLM descriptions of the two polar opposite position between which the text should be scaled. This is impractical for my use case because I want to scale Bundestag speeches that are assigned to the same topic by the topic modeling done earlier, but the topic description of those topic modeling methods only consists of a description of the topic in some form and no extreme positions. This makes Wordfish a better fit for my use case. Also, Wordfish does not fail to estimate some documents like the LLM estimators do.

Besides those practical reasons why Wordfish is more feasible than the LLMs for my use case, it is intriguing that Wordfish, the oldest method I tested, that has no sense for context at all, outperforms even the most recent method of using LLMs for that task. The LLMs I tested are mid-sized models that are way less performant than the huge models like GPT4.0 (OpenAI et al., 2024). But those same smaller LLMs performed among the best for the topic modeling task, outperforming LDA, which has the same limitations regarding context as Wordfish, by a big margin. I think it can be said that estimating the ideological position of a text is harder than categorizing it into one of 29 topics. As for the categorization of texts mainly the text is relevant but scaling the text relative to other texts, as this task requires, is harder for the LLM that sees only one text at a time. Also, there is no widely accepted definition of ideological left and right and although I provide a definition, the reason that there is no such widely accepted definition is that those terms are mostly used intuitively and who and what is branded as right or left is often contested. This vagueness certainly also is reflected in the text base the LLMs were trained on and that might make it harder for an LLM to estimate a left-right position. To address this problem, instead of just defining the extreme positions in the prompt as I did. it could work better to define more nuanced positions in the prompt and ask the LLM to assign the text to one of those positions. This would turn the scaling task into a classifying task that might be easier for the LLMs. However, this would require even more a priori knowledge of the dimension along which the texts are scaled than before.

As mentioned already Wordfish works generally very well on the manifestos with most estimations being somewhat reflective of the gold position. but Appendix Figure A.1 shows clearly that it failed to estimate the AfD party well that is put on the extreme right by the CHES survey but located in the center by Wordfish. I think this illustrates the problem of changing word meanings mentioned by Slapin and Proksch, 2008. The AfD was founded in 2013 and branded itself as the opposite to all other established parties. To express this opposition, the AfD manifestos use words in a negative context that appeared in a positive context in the majority of other manifestos. An example for this would be that most center parties were quite pro-European probably mentioning the EU and its institutions in a positive context, while the AfD then also mentioned the EU and its institutions but to express their opposition to them. So while prior to the AfD's foundation mentioning the EU a lot was indicative of being a center party, after the foundation of the AfD this changed. Wordfish cannot pick up this changing of meaning and therefore might fail to estimate the position of the AfD as well. I would expect this problem to become smaller in the future as the other parties adapt to the AfD using the words they use in a positive context with a negative connotation. An example for this would be how "Flüchtlinge" (refugees) was used by left leaning parties to express their support for refugees and after the AfD and other right wing parties started to polemize against refugees left leaning parties started to coin the term "Geflüchtete" (refugees) thereby avoiding using the old term that was given a negative connotation by the AfD. But in general, this problem of changing Word meaning over time or between different political actors remains a problem for Wordfish and needs to be kept in mind when using it. Wordfish scored the best in all scores among all tested approaches. An explanation for this is that party manifestos are usually written by the parties with great attention to detail, because they are trying to convince voters. This means they choose every word very carefully, avoiding ambiguity and preferably pick expressive words. This plays to the strength of Wordfish because Wordfish bases its estimation on the different word frequencies. Appendix Table A.2 backs this. It shows the words that differentiate most between the two extremes. These words are indeed very expressive. On the left there are words like 'gesundheitsversicherung' (health insurance), 'antifaschistische' (antifacist) and kuba (cuba) all classically associated

with left wing thinking. While on the other end there are words like 'steinkohlesubventionen' (subsidies for stone coal) 'beitragsstabilität' (Stable contributions) more associated with right wing thinking. This reasoning that can be done based on the words found by Wordfish illustrates an advantage of Wordfish. This advantage is that the results of the estimation can be validated by understanding on what grounds they were made.

The Embscal method, I decided to test, did not estimate sufficiently well that using it could be considered. The most important axis in the embedding space of all documents that should be found by the dimensionality reduction does not seem to correlate with the ideological left-right dimension. However, it can be observed in Appendix Figure A.4 that Embscal often estimates the position of the same party to be adjacent or at least close together. This probably is the reason why Embscal performed significantly better than random. But this adjacency might very well be the result of similar structure in manifestos of the same party which results in the documents being closer together in the embedding space and not an indication of Embscal ability to identify similar political positions between the documents. On the other hand Rashed et al., 2021 used Embeddings to cluster tweets into groups of similar positions, so using embeddings to cluster stances rather than scale works well.

The Glavaš method worked better than embscal but did not outperform the older Wordfish approach. This is as contraintuitive as the defeat of the LLMs against Wordfish, because Glavaš method with the sentence embeddings has a sense of context unlike Wordfish. Also even the original version tested by Glavaš et al., 2017, that used static word embeddings, outperformed Wordfish. In the original paper Glavaš method reached a perfomance level similar to the level I found on my dataset. But Wordfish scored worse on the dataset used by Glavaš et al., 2017. So the reason why Glavaš method performed worse than Wordfish on my dataset, might have more to do with how well Wordfish performed on my dataset.

4 Visualization

4.1 Goal

After identifying approaches to both the topic modeling and the stance detection, the question remains how to integrate them into the existing web application. The extracted topics need to be shown to the user, and the stances of both speakers and parties on those topics should be conveyed to the user, by the visualization. The user should be able to compare the stances of different parties and MPs. Politics is a sensitive subject and therefore it is important to make the analysis transparent. To achieve this the user should always have a fast way to read the text the machine models worked on for themself, if they are in doubt about the analysis done by the machine.

4.2 User View

In this section I will show how a user will see the improvements of the web application from topic modeling and stance detection. Whenever the name of a MP is shown in the web application the user is able to click on that name to reach a opverview page for that MP (Figure 1.4 and Figure 1.5 show this page before the increment). Now there exists a new section on each MP's page where the topics that were assigned to the speeches of the MP by BERTopic are shown. Figure 4.1 shows this topic view for the MP Ralph Lenkert. On the left side the list of topics is shown ordered by the number of speeches the MP gave on that topic. This number can be seen in the blue circle after the topic's title. If the user clicks on a topic from the list to the right of the topic list all speeches the MP gave on that topic are listed with the date they were given. The items in this speech list are clickable and lead to the page for the single speech shown in Figure 1.3. This addition the the speaker page provides users with the information which topics a MP covers, and allows users to find speeches by the speaker on topics they are interested in fast.

The second page where the user can find a new section is the page for single agenda item. The new section there shows the estimated positions of all parties on this agenda item. This section is shown in Figure 4.2. There is a vertical bar chart showing the positions of the different parties as estimated by Wordfish. the position of each bar corresponds to the position of the party. Each party is on a seperate row with the color that the party is assigned by the Bundestag. To understand what the different extreme positions mean the user can look at the list of words above the vertical bar chart. There the Wordfish found most indicative of the extreme position are listed. This helps the user to understand how the different parties view the discussed agenda item.

Those are the places where the new approaches were integrated into existing pages. The user can also view the extracted topics under a separate page. There each topic is listed both with the keywords extracted by BERTopic and with an name assigned by an LLM based on those keywords. The list of topics for the 20th election cycle of the german Bundestag can be found in

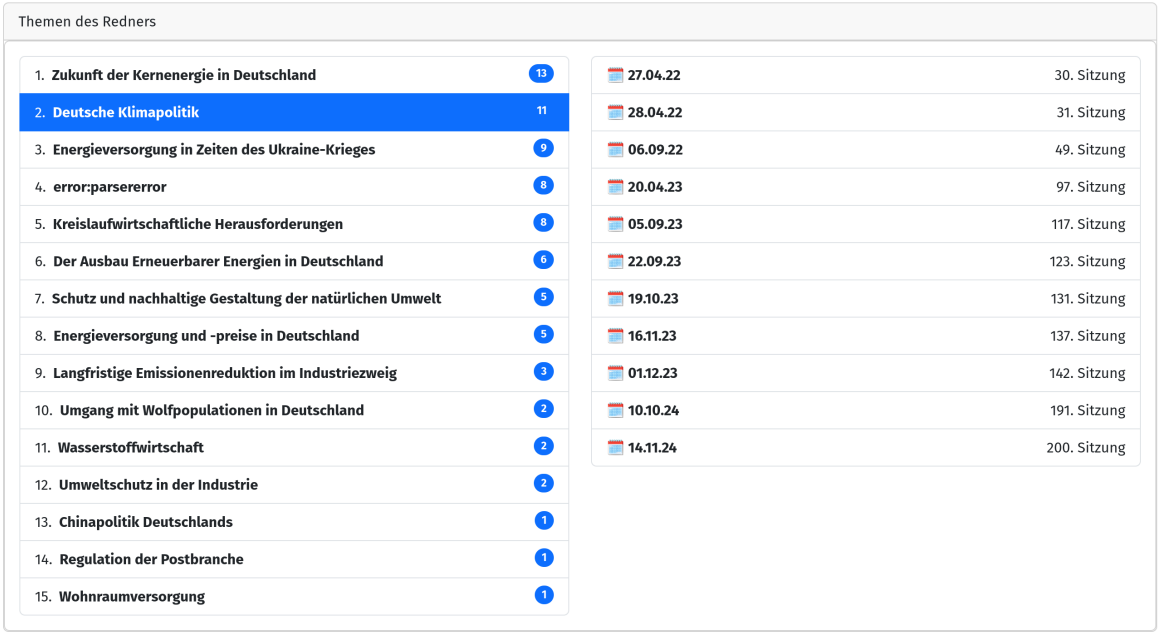

Figure 4.1: The topic List for the MP Ralph Lenkert showing on which topics he spoke.

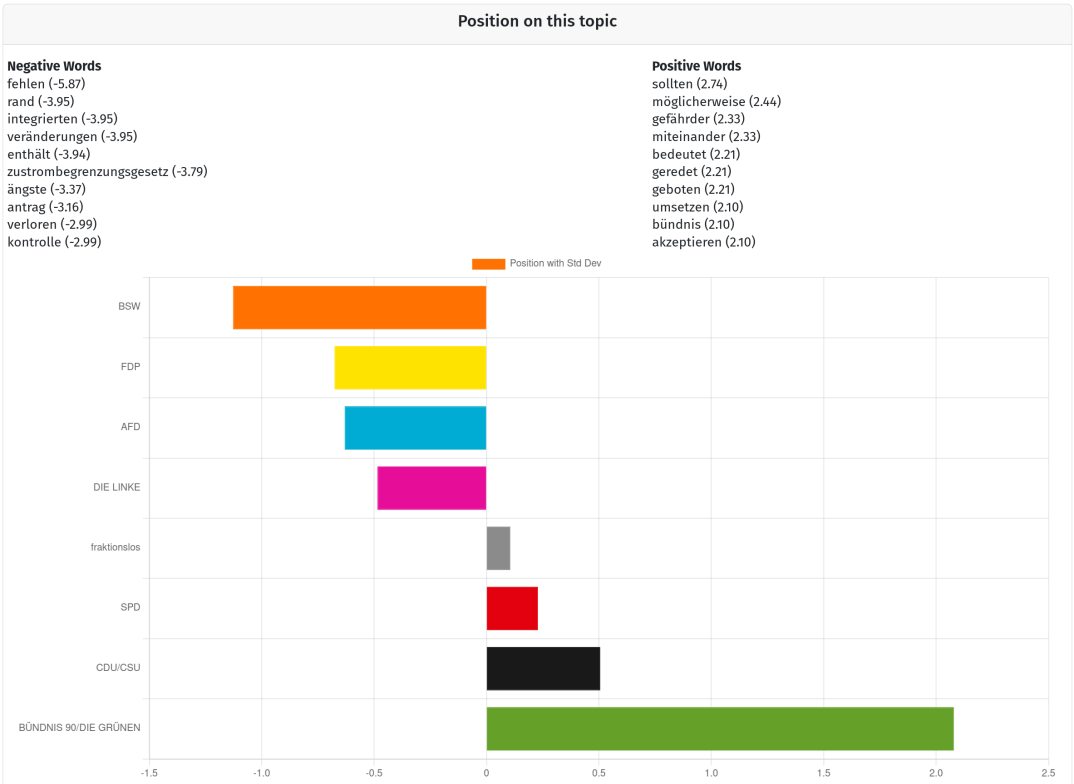

Figure 4.2: The new stance section on the agenda item overview page. First, the relevant words found by Wordfish are shown (in parenthesis behind each word is the β value of it). Then, the positions of the parties are visualized with a vertical bar chart.

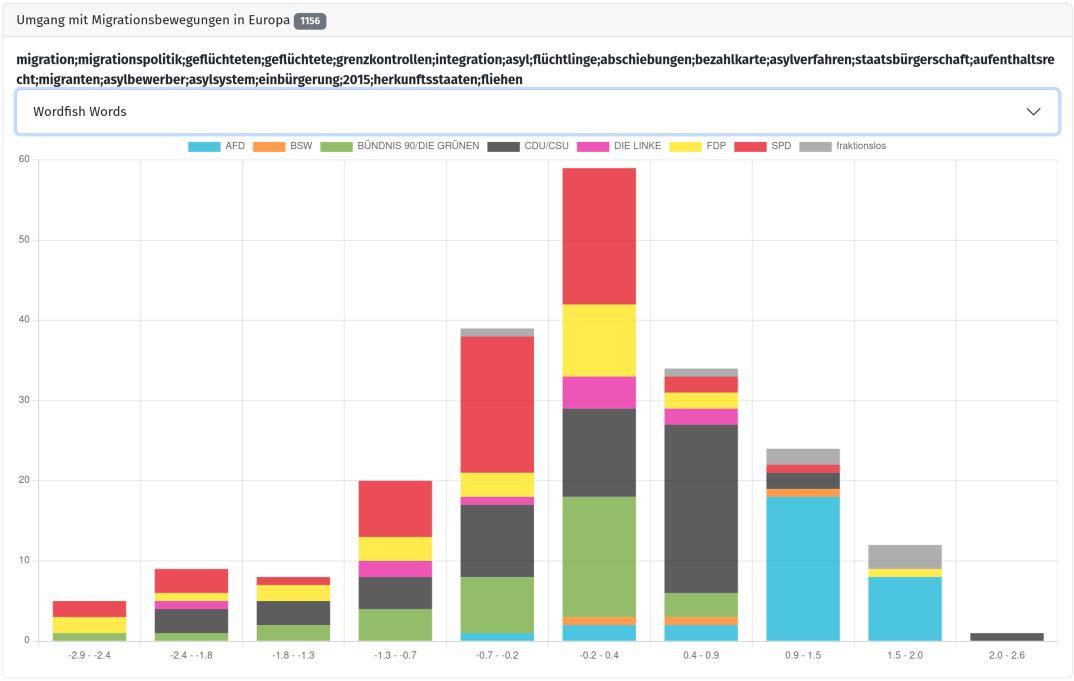

Figure 4.3: One list item in the topic list. The name and the keywords are on the top. Below that is the stacked bar chart that shows the position of different speakers on the topic

Appendix Table A.4. Each topic is listed as shown in Figure 4.3. In the header of each card is the Title of the Topic, that was assigned by an LLM. Next to the title the number of speeches assigned to that topic is shown. Below the title the most representative words for that topic are listed, to allow the user to verify the title the LLM gave the topic based on those keywords. This can be used to get an overview over all topics discussed in the German Bundestag during the 20th election cycle.

Below the topic title are information about the stances on that topic of different political actors. There are three different views for the stance analyisis available to the user. By default the users sees the positions of the speakers on that topic. Their positions are visualized as a stacked bar chart that can be seen in Figure 4.3. On the x-axis there are the positions on the topic broken down into buckets fo equal range. Like for a histogram the height of the bar shows how many speakers fall into the positions range shown on the x-axis. Each bar is divided into colored sections representing the parties of which the speakers are members. The size of the colored parts is proportional to the number of speakers from that party who were estimated in the position range of the bar. The user can also view the positions of single speeches in a stacked bar chart that works identically to the one where the positions of speakers is shown. This visualization is intended to show the user how the different positions are distributed. So for example they could see that one party has a wide range of positions among its speakers while another party's speaker are more aligned. Also to help the user understand what was deemed to be the most relevant axis, the user can uncover the words Wordfish found most relevant for the estimations. This is important because otherwise there would be no indication for the user what the different ends of the position axis actually mean.

To fulfill the goal of allowing the user to judge the position scaling for themself, they can click on any party's bar in the stacked bar chart. This brings them on a new page shown in Figure 4.4.

Figure 4.4: The overview over all speakers within the chosen position range by the chosen Party

There each speaker who was put in this position range by Wordfish is listed. For each speaker there is an expandable list where the speeches from this speaker assigned to the chosen topic can be seen. From this list the page for each speech is reachable (Figure 1.3). Above this list a Wordcloud is shown. In it only words that are among the top 400 words Wordfish found to be impactful for the scaling are shown. The size of the Word shows how often the speaker used the word over all speeches. The color indicates if the word is indicative of a scaling in the positive (red) or negative (blue) end of the scale found by Wordfish. This Wordcloud helps understand the user on what basis Wordfish scaled the speeches. For the party positions it is possible to show the relevant words in the same way as for the speeches and speaker view.

The user can view the positions of the whole parties in a seperate view. For this the stacked bar chart visualization does not work well as the height of the bars always would be one. For this reason they position there get visualized as a horizontal bar chart where the width of the bar shows the position computed by Wordfish. This visualization can be seen in Figure 4.5 and is similar to the visualization for the parties' positions on the agenda items (Figure 4.2) This can give the user an understanding of the position of the parties in relation to each other.

4.3 Technical Perspective

The technical side of the application consists of two parts. First, the analysis consisting of topic modeling and stance detection is manually run by multiple scripts. Second, the application runs continuously to keep the application reachable for users. Figure 4.6 gives an overview of how the whole system works.

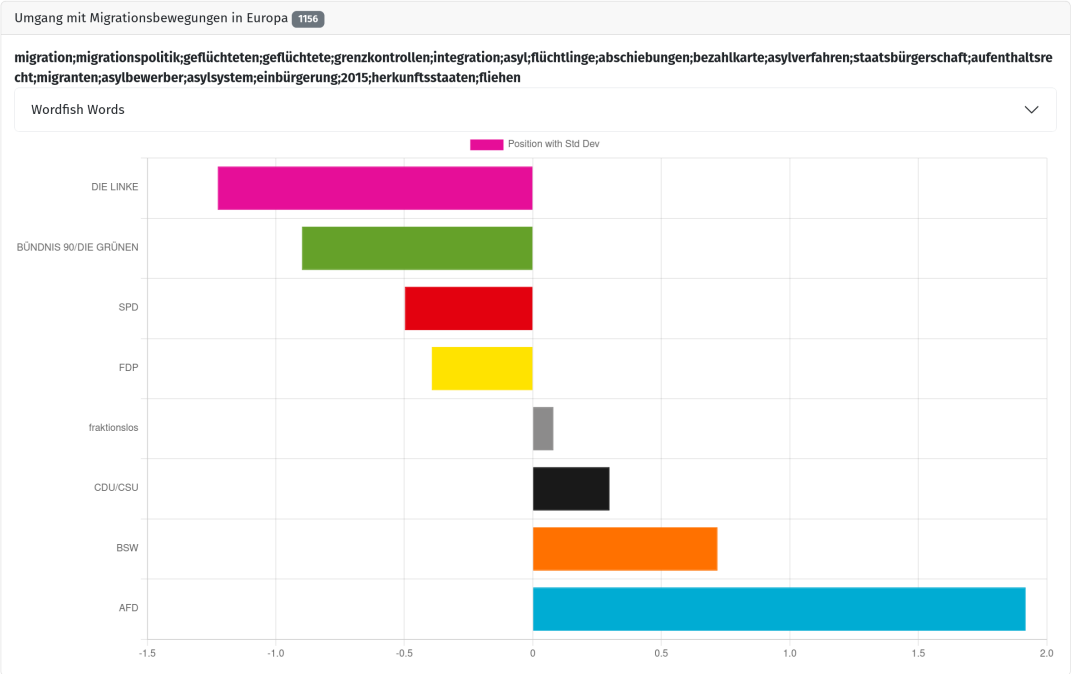

Figure 4.5: The Wordfish analysis by party for the topic migration

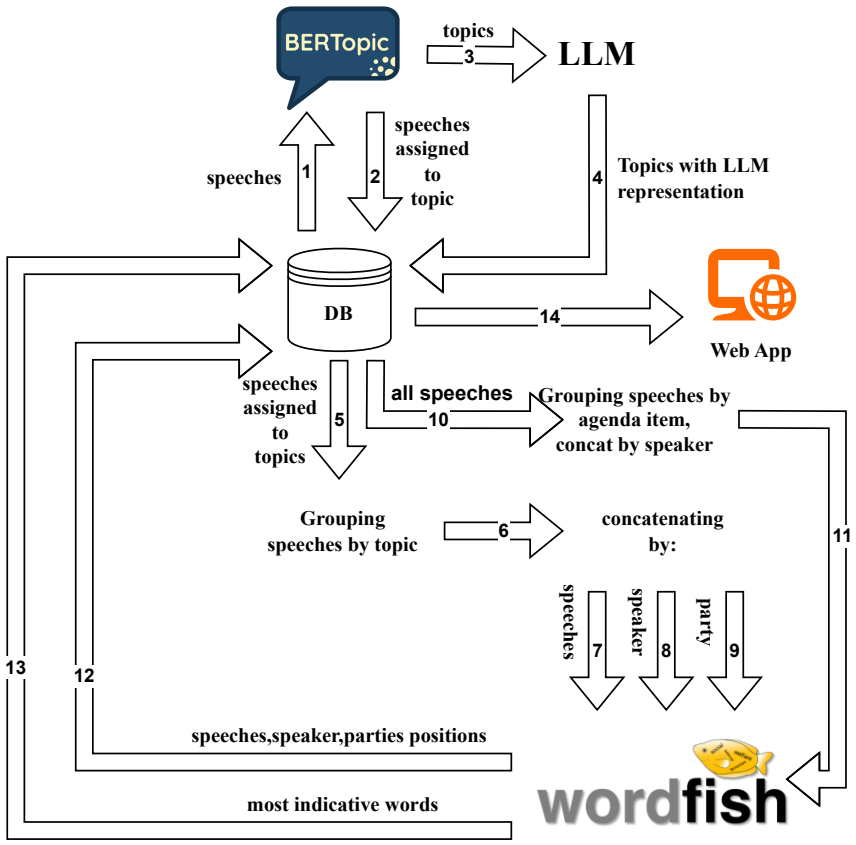

Figure 4.6: The technical process for running BERTopic and Wordfish on the Bundestag speeches

Pre-processing: Figure 4.6 contains step numbers on the arrows. Whenever a step with a number is mentioned in this section it refers to the numbers in Figure 4.6. The database that is used for the already existing application already contains the relevant data to begin the Analysis. In a first step the speches get queried from the database (Step 1). There are about 24 000 speeches from the 20th election cycle of the German Bundestag. BERTopic is run on those speeches with the jina-embeddings-v3 (Sturua et al., 2024) because they performed best during the evaluation of the topic modeling methods. To avoid the problem that Wordfish finds an axis that is not based on policy but based more a division between two topics wrongly clustered together, BERTopic was run without reducing the number of found topics afterwards. This resulted in 123 topics being found. In step 2 the assignments of speeches is saved in the database. Step 3 is prompting a LLM with the representation for each topic to obtain a title for each Topic that is clearly understandable. The representation consists of the 20 keywords found by BERTopic for each topic. *Gemma2* is used as LLM the complete prompt template can be seen in Appendix Listing A.2. In step 4 the title and the representation are saved in the database as well.

After that the topic modeling is concluded. In step 5 all speeches with their topic assignment get queried from the database. Step 6 is grouping the speeches by assigned topic. To show the user position estimations for speeches, speakers and whole parties sepratly, three seprate Wordfish runs are done for each topic. Once with the individual speeches (step 7). Once with all speeches from the same speaker concatenated (step 8). Lastly, once with all speeches from the same party concatenated (step 9). For the postions on each agenda item needed for the section shown in Figure 4.2 all speeches are grouped by agenda item and concatenated by speaker to run Wordfish once more (step 11). Then, the estimated positions are saved into the database. Also the words Wordfish deemed to be most relvant for the estimation are saved in the database.

Production The web applications backend is build with fastapi[1]. Whenever a user requests a page the relevant data is queried from the database using sqlalchemy[2]. For the topic overview page the topic assignments of all speeches and the position estimations of all speeches get queried from the database together. The fastapi backend then calculates the heigt of the different bar charts by grouping the speeches together either by speaker party or speech. Then this is given to the javascript library for chart visualization char.js[3] via the templating engine Jinja[4]. If the user clicks through to the individual speeches (Figure 4.4) all relevant Words from Wordfish for the clicked topic get queried from the database as well as all texts of the speeches in the clicked part of the bar chart. From this texts and the relevant Words the Wordcloud is computed and send to the frontend to be visualized with the d3-cloud plugin [5] for javascript. The Web pages get structured using bootstrap[6].

1. https://fastapi.tiangolo.com/ accessed March 9, 2025
2. https://www.sqlalchemy.org/ accessed March 9, 2025
3. https://www.chartjs.org/ accessed March 10, 2025
4. https://jinja.palletsprojects.com/en/stable/ accessed March 10, 2025
5. https://github.com/jasondavies/d3-cloud accessed March 10, 2025
6. https://getbootstrap.com/ accessed March 9, 2025

4.4 Discussion

I did not have the resources to quantitatively evaluate the interface, by doing a survey. Also I do not have speeches were the positions are labeled, so that I could evaluate how well Wordfish worked on the real data. For those reasons I will discuss the quality of both the machine analysis and the visualization of it based on my domain knowledge of german politics.

Party view The positions presented for the different parties in the parties overview (see Figure 4.5), gives an idea of how the parties stand relative to each other on the issue. Wordfish returns position estimates, but the user has to guess what the position axis represents. The words Wordfish returned as most expressive for each end of the position axis can help decide what the axis stands for. In the Future Work section a possible solution to that problem using LLMs is discussed briefly. Sometimes the axis Wordfish scaled the speeches along is easily understandable as for example for the topic migration. If one would guess what the most important axis of distinction between the parties on the topic migration is, probably the axis between a welcoming immigration policy and a restrictive immigration policy would come to mind. A look on the estimation from Wordfish in Figure 4.5 reveals that the Wordfish estimation correlates well with this axis. The party DIE LINKE is most associated with a welcoming immigration policy while the party AFD is deemed to have the most restrictive isolationist migration policy. Those two parties are consequently put on opposing ends of the axis by Wordfish. The Worfish estimation of the parties in between those two parties also correlate well to the position I would assign to the parties. The only party whose estimation for the migration topic is questionable is the FDP. On migration the FDP is ideologically closer the the CDU/CSU than to the SPD. Nevertheless, the estimation by Wordfish makes sense, because the FDP formed a coalition with BÜNDNIS 90/DIE GRÜNE and the SPD during most of the election cycle. So the speakers from the all coalition parties were supposed to defend the consensus of the whole coalition in their speeches. Considering this it is noteworthy that Wordfish managed to estimate the relative position of the coalition parties SPD, FDP and GRÜNE to each other correctly. BÜNDNIS 90/DIE Grüne where the most migration friendly in the coalition and the FDP the most restrictive. This shows that Wordfish was able to pick up small nuances speakers used to differentiate themself from their coalition partners while usually arguing for the same overall stance. The Words Wordfish found to be indicative of either end of the scale for migration, also show that it correlates with the welcoming vs restrictive approach. Indicative for the end I deemed welcoming are words like 'menschenrechtsverletzung' (human rights violation) and 'amnesty' (probably referring to amnesty international), while the words indicative for the other end are words like 'masseneinwanderung' (mass immigration) or 'sozialstaatsmagnete' (referring to the social safety nets being pull factors). For the migration topic the Wordfish analysis of the different parties works very vell.

There are more topics where the Wordfish analysis of the parties worked well, like for Covid-policies (shown in Appendix Figure A.6) or cultural policy. Overall for almost all topics, in the analysis by party, Wordfish estimates the SPD, Greens, and FDP next to each other. This is a indication that Wordfish worked well in identifying that those parties had to communicate their consensus to the public. In contrast to this success of Wordfish there is also a frequent problem that can be observed for example in Figure 4.7. The topic in this Figure is 'Wohnraumversorgung' (housing supply). Here, the coalition parties are estimated on the one end of the spectrum, then very close to the center-left coalition the conservative CDU/CSU is estimated. and then in

the same direction as both the conservative CDU/CSU and the far right AFD the most left-leaning party DIE LINKE is estimated. In the housing supply topic DIE LINKE is in favor for government intervention into the housing market while the AFD is strongly opposed to that. In my opinion, this should be the most important axis for that topic. Wordfish does not extract an axis that correlates with this one. An explanation why this happens is that speakers from both parties often focus on the cost of living crisis and hold the government responsible for it. In that sense the estimation by Wordfish can be explained, but I still think this is a shortcoming of the estimation because the goal is to visualize the positions of the parties and not to show rhetorical alignments.

Wordfish overall often estimates the far-right AFD close to the party DIE LINKE. In some cases estimating the position of both parties close together can be justified. In the Ukraine-War both parties have similar positions for example. But Wordfish also estimates both parties closely together for topics that concern social politics (like the housing topic in Figure 4.7). This seems wrong because both parties have opposing views on those topics. So Wordfish is able to identify a division between opposition parties and governing parties, but Wordfish often estimates the opposition parties together which is not helpful to the user, because they probably already know that government and opposition are not aligned with each other. To improve the estimations it could be interesting to run Wordfish not on the speeches but on the bills an motions submitted by the different parties. This might improve the estimation of the parties' positions because the speeches of the opposition usually are similar in condemning the government, while the submitted bills should focus on the solutions that are very dissimilar between the opposition parties.

Speaker view and speeches view The position estimation for the single speakers, shown again for the migration topic (Figure 4.3), would be valuable to users, that already are informed well about the overall positions of the parties. The speaker view (Figure 4.3) could provide information about differences inside parties, and even show which speakers voice positions that are non conforming with the majority of the party. Information about this, are not as widespread as information about the overall parties' positions. The migration topic again is one of the topics were the positions estimated by Wordfish correlates well with a reasonable policy axis. Figure 4.3 shows most most of the far-right AFDs speakers on one end the CDU/CSU speakers next to them. The speakers of the other parties are more intermingled. In contrast to the party view for the same topic most speakers from DIE LINKE does not appear on the opposed end to the AFD anymore, as they did with the same text basis but concatenated together. Another problem with this view becomes apparent when looking at the most distinctive Words by Wordfish in Table 4.1 While the Words on the positive end of the axis indicate that it is associated with a hostile view of immigration, the words indicative of the negative end of the axis are associated with education and science. This could indicate that the BERTopic topic modeling failed to separate the topics of education and migration. Then, Wordfish finds the most important axis in the incoherent topic is between education and migration. This would explain both the Words in Table 4.1 and the inexpressive estimations of all parties except the AFD. This explanation could be used to discard the whole Wordfish analysis based on speakers, but I think there is a little bit more nuance to this. The web application allows to read the speeches that are the basis for the estimation. If that is done with the speeches that led to an estimation of a speakers position on the education end of the axis, it becomes clear that they are closely related to the topic of migration. Many of them discuss the integration of young migrants into the education system. So the assignment of those speeches by BERTopic is not entirely wrong, the two topics are

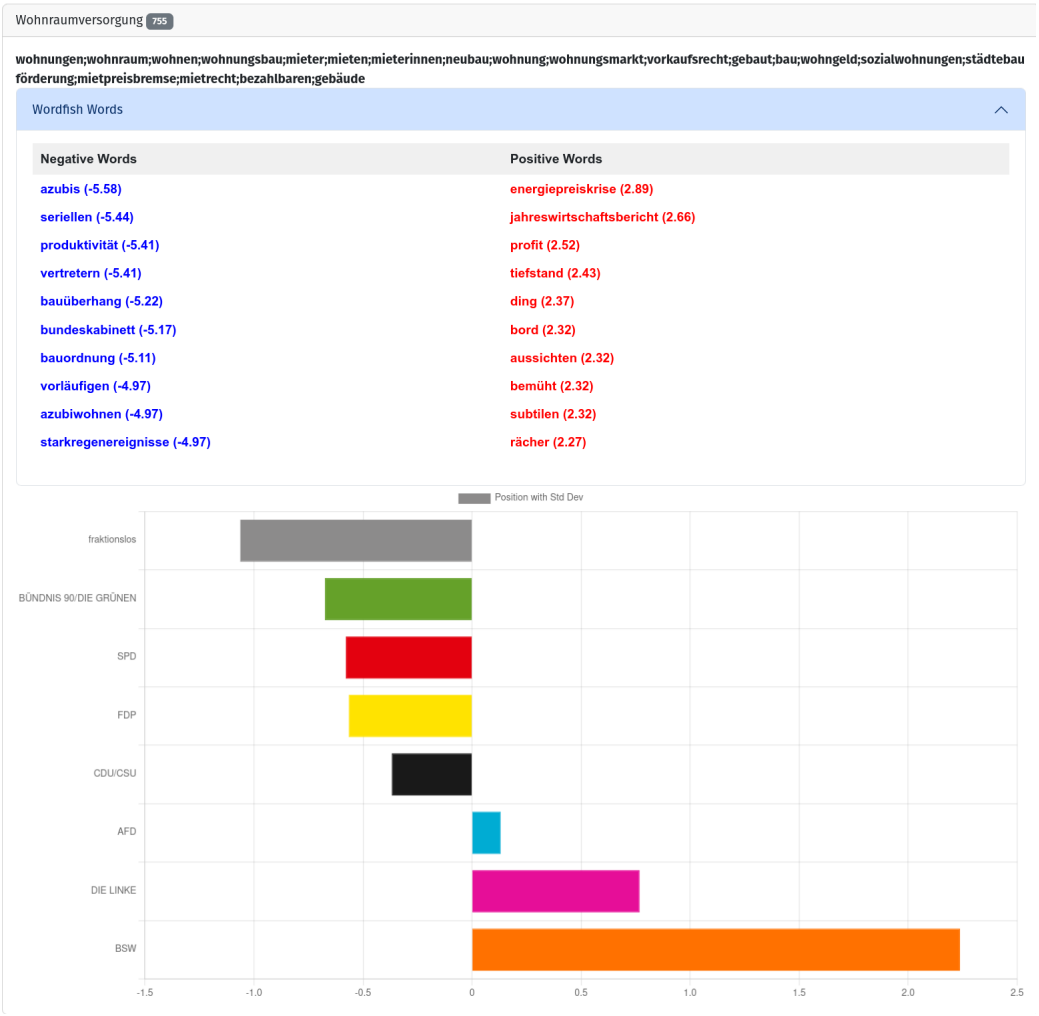

Figure 4.7: positions of the parties on the housing supply topic.

Top 10 negative Words	β	**Top 10 positive Words**	β
wissenschaftssystem	-3.00	irren	2.06
wissenschaftsfreiheit	-2.89	falschbehauptung	1.91
alexandervonhumboldtstiftung	-2.67	autobahn	1.86
ausTauschdienst	-2.65	offenkundigen	1.82
wissenschaftsdiplomatie	-2.60	entgegenkommen	1.78
akademische	-2.58	umsteuern	1.60
ludwig	-2.15	staatsgebiet	1.60
daad	-1.85	verramschung	1.60
feldzug	-1.77	import	1.60
forschungseinrichtungen	-1.77	fortschreitende	1.59

Table 4.1: The Top 10 Words Wordfish identified as placing a speaker on one end of the migration topic

closely related. Instead of an axis from pro migration to against migration Wordfish extracted an axis from a pragmatistic approach of discussing the integration of migrants into the education system, to a more fundamental discussion if migration at all is needed mainly driven by the right wing. This is an interesting finding that shows the potential of the application to offer surprising insights into the discussions in the Bundestag. That being said, the conclusion drew was not at obvious from the visualization and required reading of single speeches. Wordfish finds the axis along which it scales itself. The discussion here shows that this makes it harder to understand the scaling, because one might assume the axis should be something but Wordfish finds another axis. But Wordfish has the advantage that it can return the Words it deemed relevant for the estimations and thereby help the user to understand the extreme position between it estimated. If those labels are interpreted by the user they can understand along which axis Wordfish estimated. This can give the user a new way to look at a topic. This was not part of the goals formulated but an interesting finding.

The question remains if Wordfish can identify a spectrum of positions inside a party. My exploration suggest that this is not really something Wordfish does well in the current application. If we look again to Figure 4.3 the visualization shows that the position of two speakers from the center-left SPD is estimated in the 0.9-1.5 position range, where mostly far right AFD speakers are estimated. This would suggest that those two SPD speakers might be hardliners inside the SPD with positions close to the AFD on migration. Figure 4.8 shows that the words relevant to that estimation ('morden' (killing), 'fluten' (flood), müll ('trash') are indeed often used by the far-right when talking about migration. But when reading the speeches by those two speakers it becomes clear that both of them while using those words aim to deconstruct the far-right narrative. So putting them ideologically close to the far-right would be very wrong. This is a clear shortcoming of Wordfish. Because Wordfish works under the bag-of-words assumption and has no sense for context it is impossible for Wordfish to determine with which connotation a word is used. Because in the Bundestag speakers often address each other and try to deconstruct the arguments of the opposing side, using words that are associated with the opposing site happens very often. This problem was not as present when I evaluated Wordfish on the party manifestos because manifestos do not refer to the opposing view as often as speakers during debates in parliament do. Also when running Wordfish on the speeches concatenated by party this problem is not as prevalent, because overall a party tries to communicate their vision but single speakers sometimes mainly focus on the opposing side. But even if an approach to stance

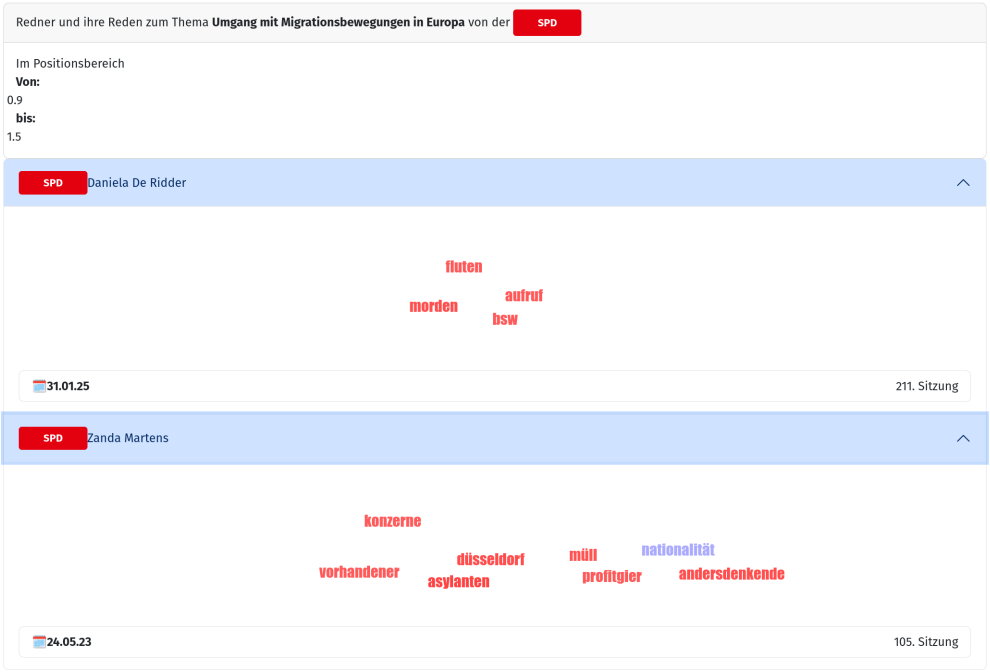

Figure 4.8: The words from the speeches of two SPD speakers that were relevant for their position estimation by Wordfish. In red are the Words suggesting a placement on the positive end of the axis. Blue are the Words suggesting a placement on the negative end

detection would account for this responsive text style of Bundestag speeches, I think extracting the spectrum of positions inside a party from them might be hard. In the German Bundestag there is a so called 'Fraktionszwang' (party discipline) that expects MPs to represent their whole party and not just their own position. It is expected of MPs that they voice their disagreements with the party line internally and not during their public speeches. So the information about different views inside one party might not be extractable from the speeches in the Bundestag alone even if an experienced expert would attempt estimating them.

The topic position view for single speeches suffers from the same shortcomings as the position estimated when grouped by speaker as Figure 4.9 shows. The stacked bar charts contain little to no useful information because the estimated speeches positions of the same party are estimated all over the the place. But sometimes interesting information can be extracted when trying to understand the axis along which Wordfish scaled.

Agenda Item View Using stance detection to find the positions of the parties on individual agenda items would be very helpful to users, who want to see what happened in a certain session without reading speeches. Also this way of stance detection on agenda items instead of topics has the advantage that it would work from the first session of a new parliament, while to extract useful topics more speeches are necessary. Assesing the performance of Wordfish when run on each agenda item seperatly for each party (Figure 4.2) is hard because there are almost 2000 agenda items in the 20th election cycle. Judging from the limited number of estiamtions I have seen for that view the estimation works poorly. While sometimes useful information are conveied via the relevant Words or the positions align with what one would expect, overall it works not well enough to provide value to the user.

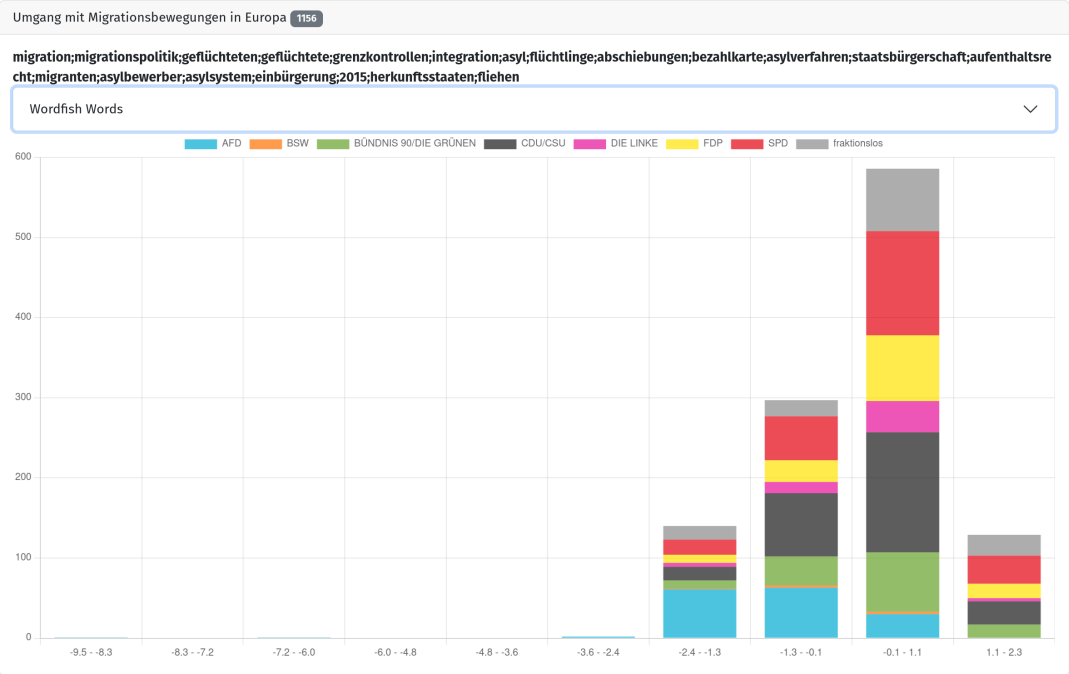

Figure 4.9: The positions for the topic migration, if Wordfish is run on individual speeches.

BERTopic Overall the topic modeling worked quite well. The list of topics a speaker covers (Figure 4.1) gives a good overview over the topics a MP covers, and makes speeches on them easily accessible. In my subjective perception, the topic overview for each speaker encourages to read more speeches because finding speech that is relevant to oneself is easy through this interface. The topic coherence seems sufficient from what I could tell after browsing through the data myself. The good topic coherence came at the cost of sometimes too fine grained topics. For example there are separate topics for most individual military deployments of the german Bundeswehr. In total there are at least 8 different topics concerning international military deployments and many more concerning the Bundeswehr in general. While it is useful for the stance detection to differentiate between the different deployments because political actors might be in favor of some but not all of them, it would be very helpful if these topics would be grouped together. Such a grouping is called topic hierarchy as there are parent and child topics. In case of the Bundeswehr there might be the Bundeswehr parent topic with child topics for domestic defense and international deployments. The international deployments child topic then would have the individual deployments as child topics. Finding such a topic hierarchy could be done with the BERTopic method of reducing topics by merging similar c-TF-IDF representations together. It could also be done with the same approach TopicGPT uses to reduce topics. This approach prompts a LLM with similar topic representations and asks it to find a name that covers both of them.

Concluding the findings of this section, It can be said that topic modeling worked very on the Bundestag speeches. This gives the user a solid structure to understand the proceedings in the Bundestag. Extracting the positions of the parties on the topics on some topics works very well, on others the Wordfish estimations mainly divided into government and opposition. Estimating individual positions of speakers or even positions of single speeches with Wordfish, did not yield useful results. Individual speakers and speeches often are to conversational or to referential to allow reliable estimation. But understanding the axis along Wordfish scaled the speakers or speeches, using the words Wordfish deemed relevant and the possibility of clicking through

the charts to the actual texts, can inspire new ways of looking at the topic. This happens because users have a preconceived notion of what the most dividing axis should be for a given topic, but when they understand the axis Wordfish found this preconceived notion is challanged. This fosters in a better understanding of the positions on the topic because a new axis of division is brought to the attention of the user.

4.5 Related Work

There are applications related to the application presented in this thesis. I will discuss some of them in this section.

Offenes Parlament The data from Offenes Parlament was already used as ground truth for the evaluation of the topic modeling in this thesis. It contains all speeches from the 18th election cycle (2013-2017) of the German Bundestag. Offenes Parlament was a project by the Open Knowledge Foundation in cooperation with *abgeordnetenwatch* [7] and *datenschule* [8] that aimed to bring more clarity to the debates in the Bundestag Offenes Parlament used their data for a web application themself [9]. They offer a full text search that can be enhanced with filters for persons, years or topic. They visualize the topics' frequencies as a bar chart (similar to Figure 2.1) and provide information about frequent speakers. Offenes Parlament did no topic modeling by machine but let annotators assign the topics. This makes their topic labels more credible compared to machine generated labels. The project is archived and does not aim to incorporate the recent election cycles.

Open Discourse Open Discourse is a non-profit project that aims to analyze all sessions of the German Bundestag. Their corpus includes all speeches between the formation of the Bundestag in 1949 and the end of the 19th election cycle in 2021. They provide a web application [10] that offeres an interactive way to extract information from this large textbase. The Application allows users to do a full text search over all speeches between 1949 and 2021. Also Open Discourse ran LDA over all speeches in their corpus. The number of topics was set to 400. After the LDA run, the number of topics was manually reduced to about 100 and useful labels were assigned by hand. The user can view the frequency of the topics over time in a line chart. This Line chart is not static but the user can modify what topics' frequencies should be shown. The user can also modifify whoose speeches should be included in the frequencies based on personal attributes like gender, age or party affiliation. This visualization is especially helpful to the user, because they can see how often a topic was discussed in a certain time period. This visualization is more advanced than just listing the topics, like my application does currently. But Open Discourse just used LDA and as this paper showed clearly LDA is outperformed by BERTopic. For this reason combining the BERTopic topic modeling with the visualization technique from Open Discourse would be a promising way to improve both applications.

7. https://abgeordnetenwatch.de accessed March 9, 2025
8. https://datenschule.de/ accessed March 9, 2025
9. https://offenesparlament.de/ accessed March 9, 2025
10. https://opendiscourse.de/ accessed March 9, 2025

Government-Opposition Divide Curini et al., 2020 used Wordfish to analyze japanese parliamentary debates between 1953-2013. In this thesis I found that the Wordfish estimation sometimes only captures the divide between government and opposition well. Curini et al., 2020 used this tendency of Wordfish to their advantage by investigating the division between government and opposition. From the Wordfish estimations for the single parties they computed and index that is inicative of the division between government and oppoisiton. The trend of this index over time allowed to identify important moments in Japans political history. Furthermore the divison index was a predictor of the time a government lasted (the lower the divsion the longer the government lasted) and how long passing of a bill took the parliament (the lower the divsion the faster bills could be passed). Adding a trend over time between government and opposition positions could improve my application because it would give users insight into how divided government and opposition are. Especially if this would be done by topic, as my application allows, it would help the user to understand the election cycle better. For example the user could see if pivotal moments like the beginning of the war in ukraine brought government and opposition closer together or tore them apart.

5 Conclusion and Future Work

5.1 Future Work

In this section I will discuss some improvements that could enhance parts of the Application and suggest some features that could be implemented in the future to provide additional value to the user.

The different stance detection approaches were evaluated based on party manifestos. While party manifestos and parliament speeches are both texts from political contexts, they do differ a lot in length and style. To pick the best stance detection approach it could be helpful to create a labeled dataset of Bundestag speeches. Then, the different stance detection approaches could be evaluated on data more similar to the real data, this might produce better insight into which approach is best suited. This also would help communicate to users how well the chosen approach works, so they can keep that in mind while viewing the results of it.

Currently the stance detection is done to differentiate between positions of different political actors (parties and MPs). Another relevant dimension for the stance detection would be time. The speeches currently get concatenated once by speaker and once by party. To analyze temporal shifts in positions the speeches could be concatenated by parliament session. This could show shifts in the overall discourse if all speeches from all parties would be concatenated together. Also concatenating speeches from the same time span and party together then could show parties positions getting closer or farther away over time. For example during the 20th election both a party (DIE LINKE) split itself into two parties and the coalition government broke apart. If the stance detection was computed over time and would work well, this breaking apart could by seen as the parties position diverge. It would be interesting from a political standpoint to see in which topics the parties diverge the most and were they stayed similar regardless of the breakup. Another way the dimension time could be included in the visualization would be to show the frequency with which topics were discussed over the whole election cycle. This visualization was already done by Open Discourse [1]. They ran LDA and visualized the Topics of all election cycles up to the 19th election cycle, but they did not analyze the 20th election period.

The visualization is based on data extracted by machine. The evaluation I did during this thesis shows that those computational methods do make mistakes. For this reason it would be helpful to allow users to report extracted information that they consider to be wrong. For example users might find some of the titles generated by the LLMs for the topics unfitting so they could propose better titles. They also should be able to flag speeches that were assigned to the wrong topic. Ideally everywhere were an information extracted by machine is shown in the interface, the user should have a report button that provides a fast way of reporting the information perceived as incorrect. This would both improve the quality of the displayed information because of the feedback from the user and work as a constant reminder to the user that the displayed information should be evaluated critically because it was extracted by a machine. Of course this would require additional resources to moderate the feedback from the users.

1. https://opendiscourse.de/diskursanalyse accessed March 9, 2025

Topic title	Negative extreme	Positive extreme
Umgang mit Migrationsbewegung in Europa	Integrationsförderer	Migrationseinschränker
Wohnraumversorgung	Herausforderungen und Missstände im Wohnungsmarkt	Strategien zur nachhaltigen Wohnraumgestaltung
Herausforderungen und Perspektiven der deutschen Landwirtschaft	Ökologisch-Integratives Modell	Marktwirtschaftlich-Protektives Modell

Table 5.1: The description of the extremes found for the three tested topics with GPT4.0

Wordfish estimates the position of the texts along a axis. To decide with to which policy axis this estimated axis correlates best, the user currently has to view the list of words extracted by Wordfish and read the speeches estimated at the extreme positions. Doing this the user can find what axis fits best and how they would describe the extreme positions of that topic. This process could be done by a LLM instead. This idea is similar to how the LLM was used to assign topic titles based on the keywords extracted with BERTopic. I tested this with three example Topics namely the topics: 'Umgang mit Migrationsbewegung in Europa' (handling of migration in europe), 'Wohnraumversorgung' (housing supply), and 'Herausforderungen und Perspektiven der deutschen Landwirtschaft' (challenges for german agriculture). I prompted GPT4.0 with a prompt asking it to assign the extremes a label. To do this I included in the prompt the 10 most expressive words extracted by Wordfish and the three speeches Wordfish estimated to be most extreme for each end of the axis. The full prompt can be seen in Appendix Listing A.4. The description the LLM assigned to the extreme positions can be seen in Table 5.1. The full answers can be read in Appendix Listing A.5, Appendix Listing A.6 and Appendix Listing A.7. The descriptions for the migration topic and the agriculture topic seem reasonable, while the description of the housing topic is quite ambigious as the one extreme is called 'Herausforderungen und Missstände im Wohnungsmarkt' (Challenges and grievances in the housing market), which is not very descriptive for a political position. I think overall this idea showed potential to address the problem of finding along which policy axis Wordfish estimated the positions.

If a user wants to compare the positions of different parties on multiple topics, they currently have to scroll between the multiple bar charts (shown in Figure 4.5) of the different topics. This scrolling between topics makes the comparison harder. To address this issue allowing the user to select the topics they want to compare themself would help. Also allowing to select only a subset of parties would help keeping the visualization simple. The positions of the selected parties on the selected topics then could be visualized in a so called radar-chart. A radar chart is a visualization method that allows to visualize many dimensions at the same time. Each dimension is visualized as a radial axis of of a polar coordinate system. An example of such a radar chart with fictive positions of fictive parties can be seen in Figure 5.1. This visualization is convenient for the user because they can select the topic they are interested in and compare multiple parties positions in one graphic.

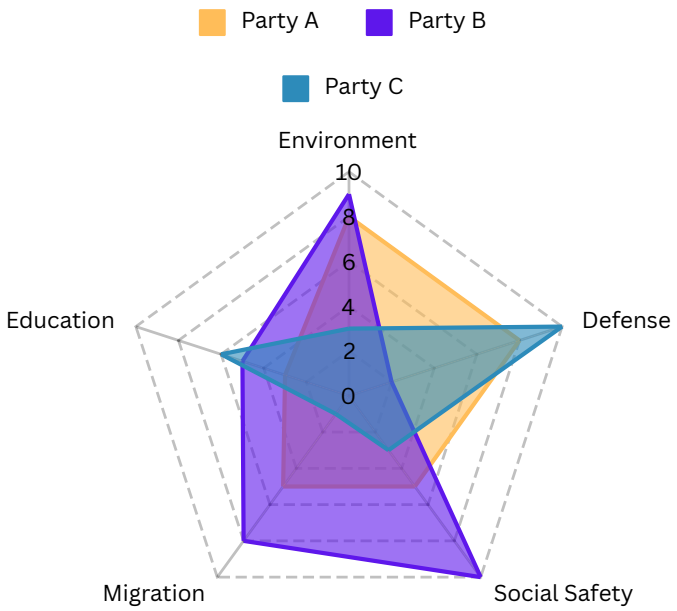

Figure 5.1: A fictive radar chart showing the positions of three parties on five topics.

5.2 Conclusion

To conclude this thesis I will look back on the findings of the previous chapters. I already created an application that provided summaries of the proceedings in the German Bundestag. This application followed the structure of the proceedings in the Bundestag. To give the user a way of understanding those proceedings through a more intuitive structure, topic modeling was chosen. Topic modeling is the task of structuring text corpora into topics that are understandable. The German Bundestag is a political institution and for this reason the political stances voiced in it are relevant. Estimation of political stances can be done through stance detection. For those reasons the thesis focused on finding approaches to both topic modeling and stance detection, that work well with the data from the German Bundestag.

I tested and evaluated different topic modeling methods in the context of German politics. More specifically I obtained Bundestag speeches where topic labels were assigned by hand to evaluate the topic modeling methods on. I tested approaches that reflect three different approaches to natural language processing as a whole. With LDA I tested an old approach using the bag of words assumption that is blind to context. A more modern approach using Sentence Embeddings was tested with BERTopic. And the most recent approach of utilizing the power of LLMs was tested with TopicGPT. Looking at the results of the evaluation, it can be said that the oldest LDA approach is outperformed by all modern approaches in all tested metrics. This shows that topic modeling performance for German political texts is greatly enhanced if a method that does not rely on the bag of words assumption is used. The testing also revealed that the performance of BERTopic can be improved by using sentence embeddings that were trained for multilingual or German clustering. The results clearly show that LLMs reach high accuracy when assigning documents to topics, but the tested mid-sized models fail to generate topics that generalize well so using TopicGPT for my application was not feasible. Overall the most feasible approach for the German Bundestag data is to use BERTopic with fitting sentence embeddings.

After topic modeling, I investigated the performance of different approaches to stance detection. Again I chose approaches that cover a wide array of natural languae processing methods. The oldest chosen approach was Wordfish that relied on the bag of Words assumption. A more recent approach was tested with Glavaš method . I attempted to improve this method by using sentence embeddings instead of static word embeddings. I also tested an approach called Embscal that relied on sentence embedding and dimensionality reduction to estimate the positions. Lastly, I tested how well LLMs could estimate ideological positions of texts. I evaluated the stance detection using a dataset of German party manifestos combined with left right estimations of those parties by experts. The Stance detection evaluation resulted in an unexpected finding. While the method relying on the bag-of-words assumption did not work well for topic modeling, Wordfish, that assumes conditional independence of all words as well, scored the best scores in almost all metrics. One reason it performed so well is that the careful chosen words, of which the used dataset of party manifestos consisted, make the estimation for Wordfish easier. The more modern approach by Glavaš et al., 2017 worked on the tested dataset but it did not achieve high scores. Using LLMs to extract stances on a scale as was necessary for my task worked well but still not better than using Wordfish for the party manifestos. The LLMs were held back by the vagueness of the left-right political classification, and the problem that they only see one text at a time. This makes estimating the relative position of these texts harder. The tested Embscal method did not achieve significantly better performance than a random estimation did and can be disqualified. Overall the approach most suited for the German party manifestos was Wordfish, so I picked Wordfish to estimate the position of the Bundestag speeches as well.

After the evaluation was done and I picked the approaches to topic modeling and stance detection that seemed most suited to achieve the goal of providing accesible information extracted from the Bundestag speeches, I build multiple visualizations for the results of both topic modeling and stance detection that was included into the existing web application. Before visualizing I had to run both BERTopic and Wordfish on the actual speeches from the 20th election cycle of the German Bundestag. The results of that were saved in a database from which the web application queries the relevant data for the visualization. First, I used this data to improve existing pages of the Web application. I added a vertical bar chart on each agenda item's page that shows the estimated position of all parties on that agenda item. While this visualization would be very useful to grasp an agenda item very fast without reading all speeches, the performance of Wordfish on the agenda items was not sufficiently reliable. Second, I added an overview over the topics each MP gave speeches on to the overview page of each MP. This provides the users, who are interested in a specific MP (for example because the MP represents their district), an easy way to sense the area of expertise of the MP. Also this view allows the user to access the full text of the speeches by the MP on each topic. Then, I created a new page where each extracted Topic is listed with the representation extracted by BERTopic and a title found by a LLM. For each topic there are three ways of visualizing the estimated positions on that topic. First the positions of the parties can be visualized in a vertical bar chart. Extracting the positions of the parties on the found topics worked well, with the caveat that often just the difference between government parties and opposition parties was found while differences between the opposition parties were not extracted sufficiently. Second, The positions of individual speakers are visualized in a stacked bar chart. This stacked bar chart has the potential of showing the spread in a party's position. Also finding speakers that dissent from the consensus of their party could be possible with this visualization. But extracting the positions of a single speaker did not result in meaningful position estimations. But the axis Wordfish finds when trying to extract the positions often give insight into interesting aspects of the topic that is discussed. The same holds

for the third way of visualizing the positions that is showing the estimation for each speech in a stacked bar chart. The visualization could be improved if the axis extracted by Wordfish was automatically labeled through prompting a LLM. Also the structure would greatly improve if the topics were organized in a topic hierarchy. The Application gives users useful information and allows them to follow the curiosity sparked through the visualization, by always making the original speeches available.

Overall the topic modeling with BERTopic on the Bundestag speeches worked solidly, while the stance detection faced some challenges. The fact that I was able to identify the challenges Wordfish encountered on the Bundestag data, shows that the visualization helps to convey the results of the stance detection to users. The improved application that results from this thesis shows the potential of computational analysis of legislative sessions. The stance detection in German legislative session of whole parties yielded promising results. For the proportional representation, created by the German election system, parties' positions are the most important to the voter, because they mainly vote for parties not individual representatives. The application improved in this thesis makes the positions of the parties more accessible to users, this allows users to inform themselves independently on topics regardless of which topics dominate the news cycle currently.

This thesis was able to show the potential of combining stance detection and topic modeling into one visualization. It became clear that interactivity and transparency are important values for applications that use natural language processing to extract information about political processes. This transperency allowed the assessment that BERTopic works very well on German Bundestag speeches, while Wordfish faces challenges that need to be improved upon to create an even better aid to users, who want to inform themselves. While the application improved through this thesis cannot replace established methods of gathering information about parliamentary proceedings, it clearly opened up a new avenue of generating insight about political discourse and party positions in the Bundestag.

Bibliography

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3 (Jan): 993–1022.

David M. Blei. 2012. Probabilistic topic models. *Commun. ACM* (New York, NY, USA) 55, no. 4 (April): 77–84.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's Next Language Model. In *Proceedings of the 28th International Conference on Computational Linguistics,* edited by Donia Scott, Nuria Bel, and Chengqing Zong, 6788–6796. Barcelona, Spain (Online): International Committee on Computational Linguistics, December.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* edited by Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, 8440–8451. Online: Association for Computational Linguistics, July.

Luigi Curini, Airo Hino, and Atsushi Osaka. 2020. The Intensity of Government–Opposition Divide as Measured through Legislative Speeches and What We Can Learn from It: Analyses of Japanese Parliamentary Debates, 1953–2013. *Government and Opposition* 55 (2): 184–201.

Roman Egger and Joanne Yu. 2022. A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts. *Frontiers in Sociology* 7.

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,* edited by Mirella Lapata, Phil Blunsom, and Alexander Koller, 688–693. Valencia, Spain: Association for Computational Linguistics, April.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and Ahmad Al-Dahle. 2024. The Llama 3 Herd of Models. arXiv: 2407.21783 [cs.AI].

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv* (March). eprint: 2203.05794.

Nils Constantin Hellwig, Jakob Fehle, Markus Bink, Thomas Schmidt, and Christian Wolff. 2024. Exploring Twitter discourse with BERTopic: topic modeling of tweets related to the major German parties during the 2021 German federal election. *International Journal of Speech Technology* 27 (4): 901–921.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, and Gianna Lengyel et al. 2023. Mistral 7B. arXiv: 2310.06825 [cs.CL].

Thorsten Joachims et al. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *ICML,* 97:143–151.

Seth Jolly, Ryan Bakker, Liesbet Hooghe, Gary Marks, Jonathan Polk, Jan Rovny, Marco Steenbergen, and Milada Anna Vachudova. 2022. Chapel Hill Expert Survey trend file, 1999–2019. *Electoral Studies* 75:102420.

Alon Lavie and Michael J Denkowski. 2009. The METEOR metric for automatic evaluation of machine translation. *Machine translation* 23:105–115.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software* 2 (11): 205.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426.*

Geoffrey J McLachlan and Thriyambakam Krishnan. 2007. The EM algorithm and extensions. John Wiley & Sons.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and Sentiment in Tweets. *ACM Trans. Internet Technol.* (New York, NY, USA) 17, no. 3 (June).

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics,* edited by Andreas Vlachos and Isabelle Augenstein, 2014–2037. Dubrovnik, Croatia: Association for Computational Linguistics, May.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics: ACL 2022,* edited by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, 1864–1874. Dublin, Ireland: Association for Computational Linguistics, May.

OpenAI et al. 2024. GPT-4 Technical Report. arXiv: 2303.08774 [cs.CL].

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. Doha, Qatar: Association for Computational Linguistics, October.

Chau Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024. TopicGPT: A Prompt-based Topic Modeling Framework. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers),* edited by Kevin Duh, Helena Gomez, and Steven Bethard, 2956–2984. Mexico City, Mexico: Association for Computational Linguistics, June.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21 (140): 1–67.

Ammar Rashed, Mucahid Kutlu, Kareem Darwish, Tamer Elsayed, and Cansın Bayrak. 2021. Embeddings-Based Clustering for Target Specific Stances: The Case of a Polarized Turkey. *Proceedings of the International AAAI Conference on Web and Social Media* 15, no. 1 (May): 537–548.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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),* edited by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, 3982–3992. Hong Kong, China: Association for Computational Linguistics, November.

Andrew Rosenberg and Julia Hirschberg. 2007. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL),* edited by Jason Eisner, 410–420. Prague, Czech Republic: Association for Computational Linguistics, June.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20:53–65.

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.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual Embeddings With Task LoRA. arXiv: 2409.10173 [cs.CL].

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9 (11): 2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems,* edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, vol. 30. Curran Associates, Inc.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report. *arXiv preprint arXiv:2402.05672.*

Silvan Wehrli, Bert Arnrich, and Christopher Irrgang. 2023. German Text Embedding Clustering Benchmark. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023),* edited by Munir Georges, Aaricia Herygers, Annemarie Friedrich, and Benjamin Roth, 187–201. Ingolstadt, Germany: Association for Computational Linguistics, September.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations.* Online.

A Appendix

Table A.4: The 123 topics BERTopic found in the Bundestag speeches of the 20th election cycle with the Title the LLM assigned to them and the number of times a speech was assigned to the topic.

LLM title	**n**
OUTLIER	6521
Umgang mit Migrationsbewegungen in Europa	1157
Wohnraumversorgung	755
Motorisierte Transportpolitik und Infrastruktur	644
Haushaltspolitik im Spannungsfeld von Klimaschutz und Ausgabenkontrolle	475
Pandemie-Politik	466
Wirtschaftsstandort und Wettbewerbsfähigkeit in Deutschland	453
Die israelisch-palästinensische Konflikt	394
Sicherheit und Kriminalität in Deutschland	384
Gegenwart und Zukunft der Krankenhausversorgung	373
Digitalisierung der Verwaltung	348
Herausforderungen und Perspektiven der deutschen Landwirtschaft	346
Armut von Kindern und Familien	325
Zuständige Ministerien und Verfahren im Umgang mit Autismus	312
Modernisierung der Bundeswehr	311
Entwicklungszusammenarbeit im 21. Jahrhundert	303
Internationale Lage und der deutsche Beitrag	303
error:parsererror	283
Sicherung des Arbeitsmarktes und Geldbestendung im Spannungsverhältnis zu grundsicherung.	279
Auswirkungen der deutschen Einheit auf Ostdeutschland	278
Reform der Altersversorgung	237
parlamentarische Transparenz und Rechenschaftspflicht	229
Deutsche Klimapolitik	208
Frühkindliche Bildungschancen und ihre Verbesserung	199

Model	**Accuracy**	**Precision**	**Recall**	**f1**
BERTopic	**19.32%**	**23.22%**	**19.32%**	**19.10**
LDA	16.72 %	21.03%	16.72%	16.50%

Table A.1: The scores if the assigned Topics get removed from the prompts for the LLM representation mapping

LLM title	n
Datenschutz und Strafverfolgung	195
Energieversorgung in Zeiten des Ukraine-Krieges	194
Zukunft der Kernenergie in Deutschland	192
Militärhilfe für die Ukraine	188
Pflege im Gesundheitssystem	188
Bekämpfung der Inflation und soziale Entlastung	180
Gewalt gegen Frauen	180
Die Zukunft der Automobilindustrie	175
Naturkatastrophenvorsorge und -resilienz	170
Kompetenzen für die Zukunft	164
Qualität und finanzielle Sicherung der frühkindlichen Bildung	163
Abbau von Verwaltungsverfahren	146
Forschung und Wissenschaftsförderung in Deutschland	145
Der Einsatz der Bundeswehr in Afghanistan und Syrien	142
Reform des Deutschlandtickets für ÖPNV	141
Förderung des Sports in Deutschland	141
Cannabispolitik	139
Finanzierung von Ausbildung und Studium	136
Internationale Handelsabkommen	134
Inklusion von Menschen mit Behinderungen	130
Europäische Beteiligung an der Krisenbewältigung in Libyen	129
Sicherheit der Fachkräfteeinwanderung	129
Digitale Rechtssprechung und Verbraucherrechte	125
Einsatz der Bundeswehr in der Sahelregion	124
Kreislaufwirtschaftliche Herausforderungen	115
Nachhaltige Gestaltung der Zukunft	115
error:parsererror	113
Umgang mit Wolfpopulationen in Deutschland	111
Nachhaltige Tierhaltung	111
Kulturpolitik	107
Bundespolitik in Zeiten der Krise	107
Aufgaben und Herausforderungen der Bundeswehr in Zeiten des Wandels	106
Globale Ernährungssicherheit	103
Iran im Fokus der internationalen Politik	101
Bedürfnisse von Menschen mit trans- und intersexueller Identität	100
Schutz und nachhaltige Gestaltung der natürlichen Umwelt	98
Bafög und Bildungssystem in der Digitalisierung	98
Islamismus und Integration	96
Energieversorgung und -preise in Deutschland	96
Justizdiskussion	96
Chinapolitik Deutschlands	93
Rechtliche und gesellschaftliche Rahmenbedingungen von Schwangerschaftsabbrüchen in Deutschland	93
Mitbestimmung in der Arbeitswelt	88
Der Ausbau Erneuerbarer Energien in Deutschland	87

LLM title	n
Regulierung und gesellschaftlicher Umgang mit Künstlicher Intelligenz	86
Die Zukunft der Westbalkanregion	83
Arzneimittelversorgung	82
Datenschutz in der Digitalisierung	78
Völkerstrafrecht und Kriegsrechable	78
Langzeitfolgen von COVID-19	77
Ernährungspolitik und zivilgesellschaftliche Beteiligung	77
Cum-Ex-Skandal in Hamburg	76
Auswirkungen der Corona-Pandemie auf die Gastronomie	74
Das Grundgesetz und der Rechtsstaat	72
Unterricht und Kontrolle des Bundespolizei-Einsatzes	72
Beitrag der Bundeswehr im Irak	72
Die Zukunft der deutschen Schifffahrt	71
Internationale Friedensmissionen in Südsudan	71
Schutz kritischer Infrastrukturen vor Cyberangriffen	70
Wahlsystemreform	69
Wasserstoffwirtschaft	68
Gestaltung der Arbeitszeitmodelle in der Zukunft	67
Die Menschenrechtslage in Lateinamerika und der Rolle der internationalen Staatengemeinschaft.	66
Bürgerbeteiligung im Gesetzgebungsprozess	66
Regulierung der Medienlandschaft	66
Sterbebegleitung und Autonomie am Lebensende	65
Bekämpfen der Finanzkriminalität	63
Regulation der Postbranche	63
Die Bedeutung von Vereinen für die Gesellschaft	62
error:parsererror	62
Politische Proteste im Zusammenhang mit Klimaschutz	60
Globale Besteuerung von Multinationale Unternehmen	60
Reform des Solidaritätszuschlags	60
Digitale Transformation im Gesundheitswesen	56
Die UNIFIL-Mission in Libanon 2006	56
Auswirkungen der COVID-19 Pandemie auf die psychische Gesundheit von Kindern und Jugendlichen	56
Der Euro und seine Folgen	56
Erinnerungskultur zum Nationalsozialismus	55
Die Aktualisierung der deutschen Sicherheitspolitik	55
Deutsche Beteiligung an der Friedenssicherung im Kosovo	55
Sicherheitspolitik im Innenbereich	54
Die Entwicklung des Luftverkehrs in Deutschland	52
Arbeitsbedingungen in der Tarifverhandlung	52
Anpassung des Steuerrechts an die Folgen der COVID-19-Pandemie	52
Deutsch-französische Beziehungen	51
Verbraucherrecht und -schutz in der digitalen Welt	51
Politische und Rechtsstaatliche Maßnahmen gegen russische Oligarchen	51
Europäische Kooperation im Rahmen globaler Herausforderungen	49

LLM title	n
Regulierung und Zukunft des Bankensektors	48
error:parsererror	48
Regulierung von Online-Plattformen	46
Regulierung von Lieferketten	44
Förderung von Innovationen aus Wissenschaft und Forschung	44
Zukunft der Pflanzenschutzmittel in der EU	44
Umweltschutz in der Industrie	43
Internationale Koordinierung der Pandemiebekämpfung	43
Langfristige Emissionenreduktion im Industriezweig	43
Reform des Wissenschaftszeitvertragsrechts	42
Reform des deutschen Wahlrechts zum unmittelbaren Wählen von Abgeordneten im Europaparlament	40

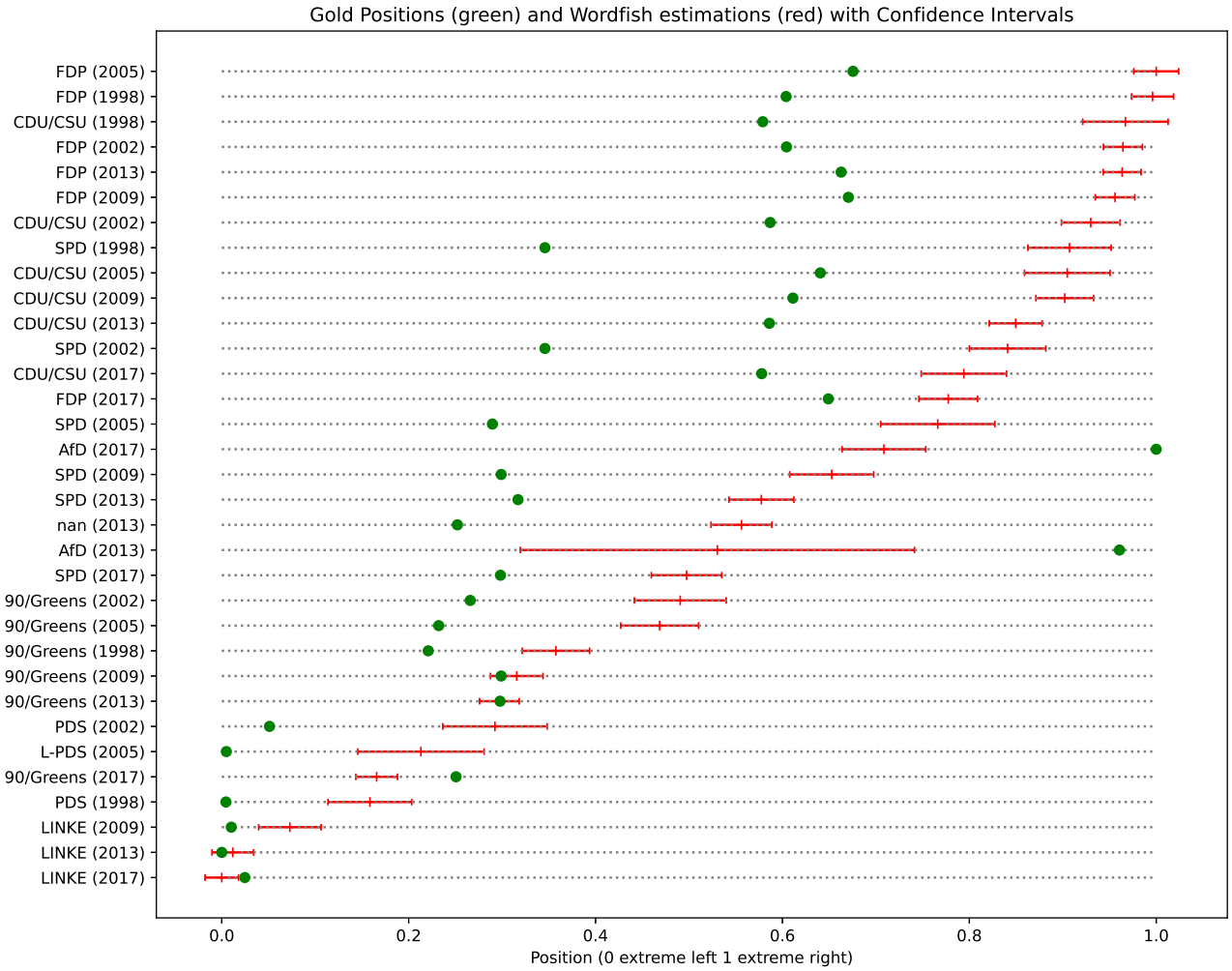

Figure A.1: The Wordfish estimations of all parties compared to the gold positions of the manifestos. In parantheses is the election year to which the manifesto was published. both positions are min-max scaled to be between 0 and 1. If there is no estimation point and the line is not dotted

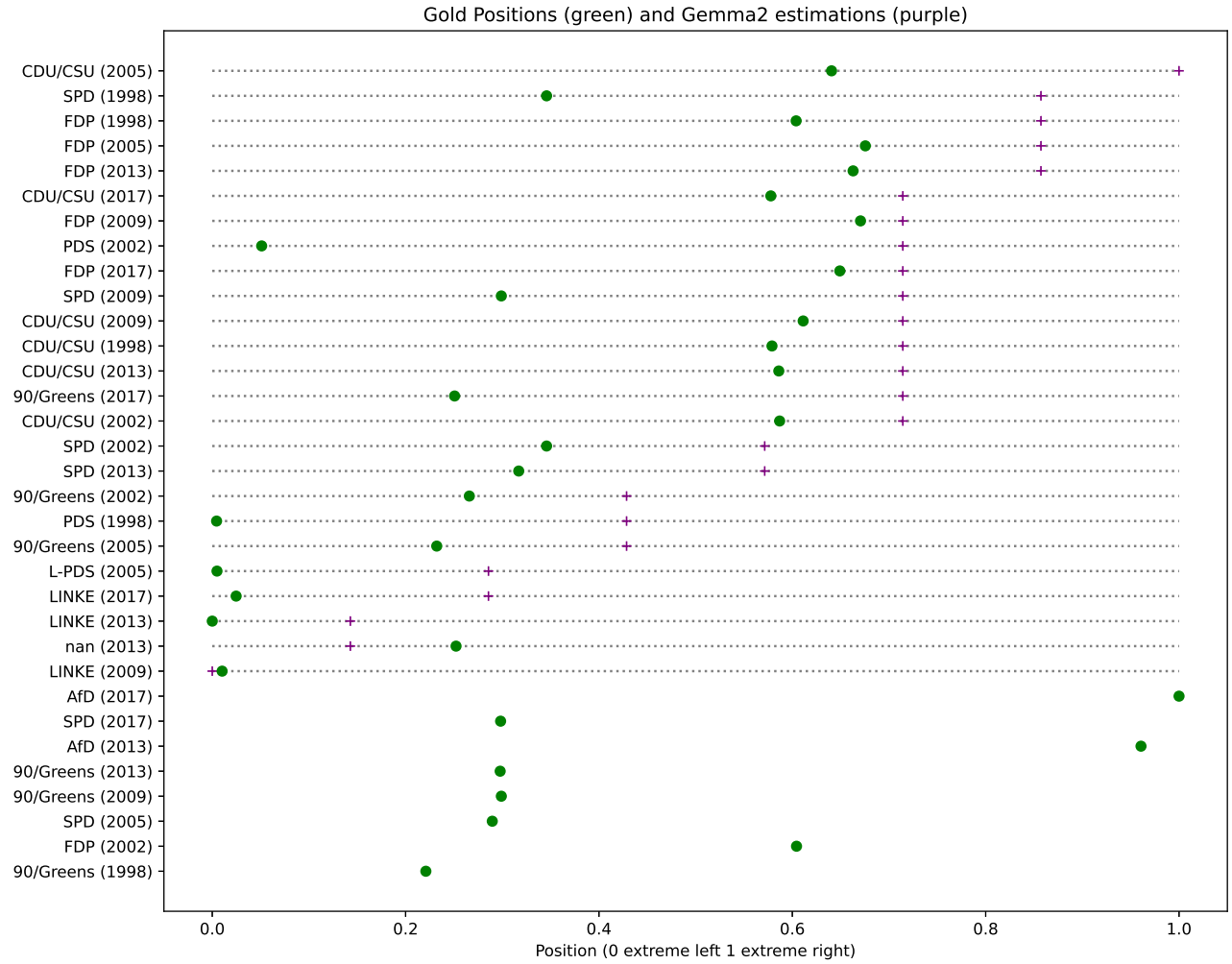

Figure A.2: Gemma2 estimations compared to the gold positions of the manifestos from different parties. In parantheses is the election year to which the manifesto was published. both positions are min-max scaled to be between 0 and 1. If there is no estimation point and the line is not dotted Gemma2 failed to scale this Manifesto

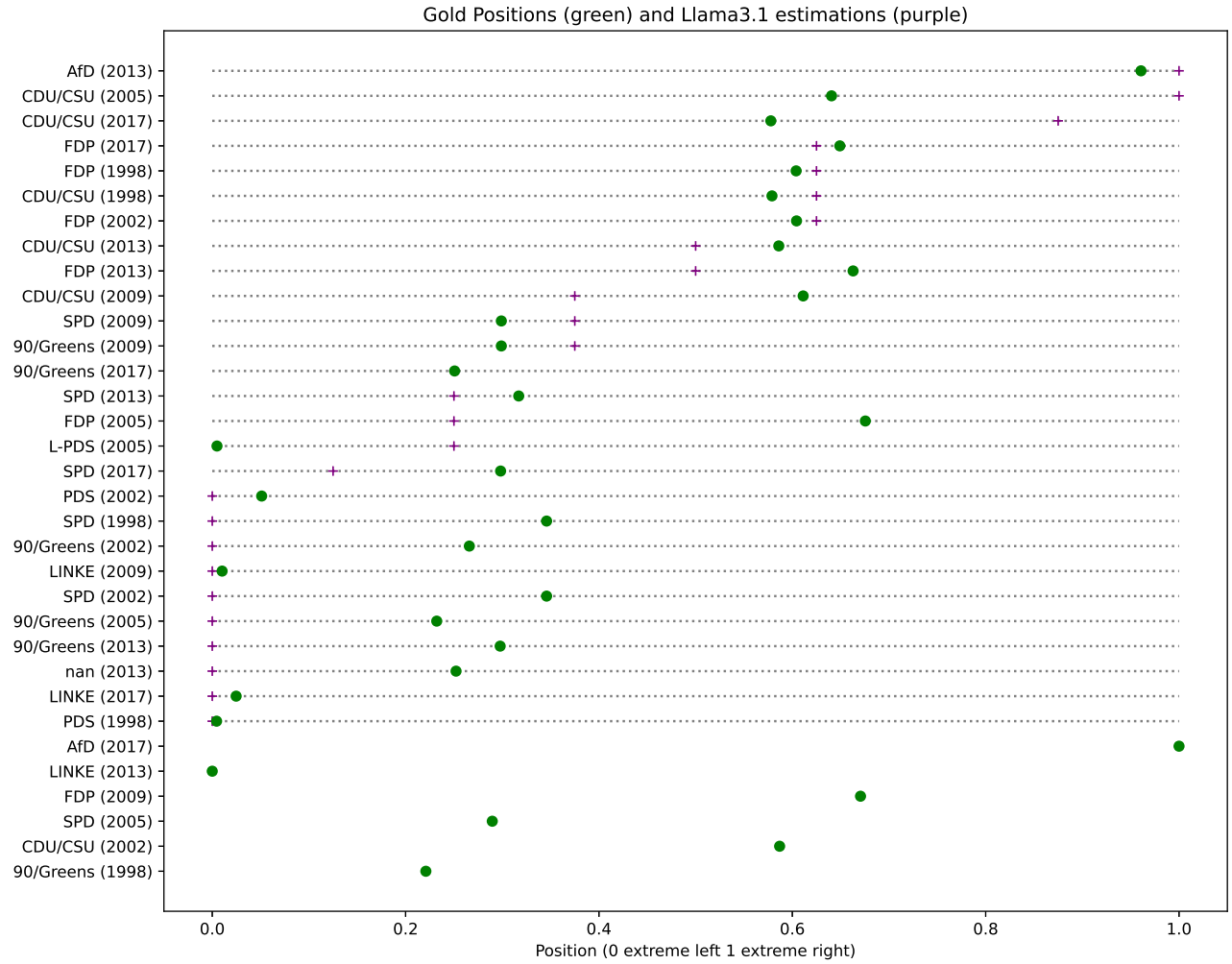

Figure A.3: Llama3.1 estimations compared to the gold positions of the manifestos from different parties. In parantheses is the election year to which the manifesto was published. both positions are min-max scaled to be between 0 and 1. If there is no estimation point and the line is not dotted Llama3.1 failed to scale this Manifesto

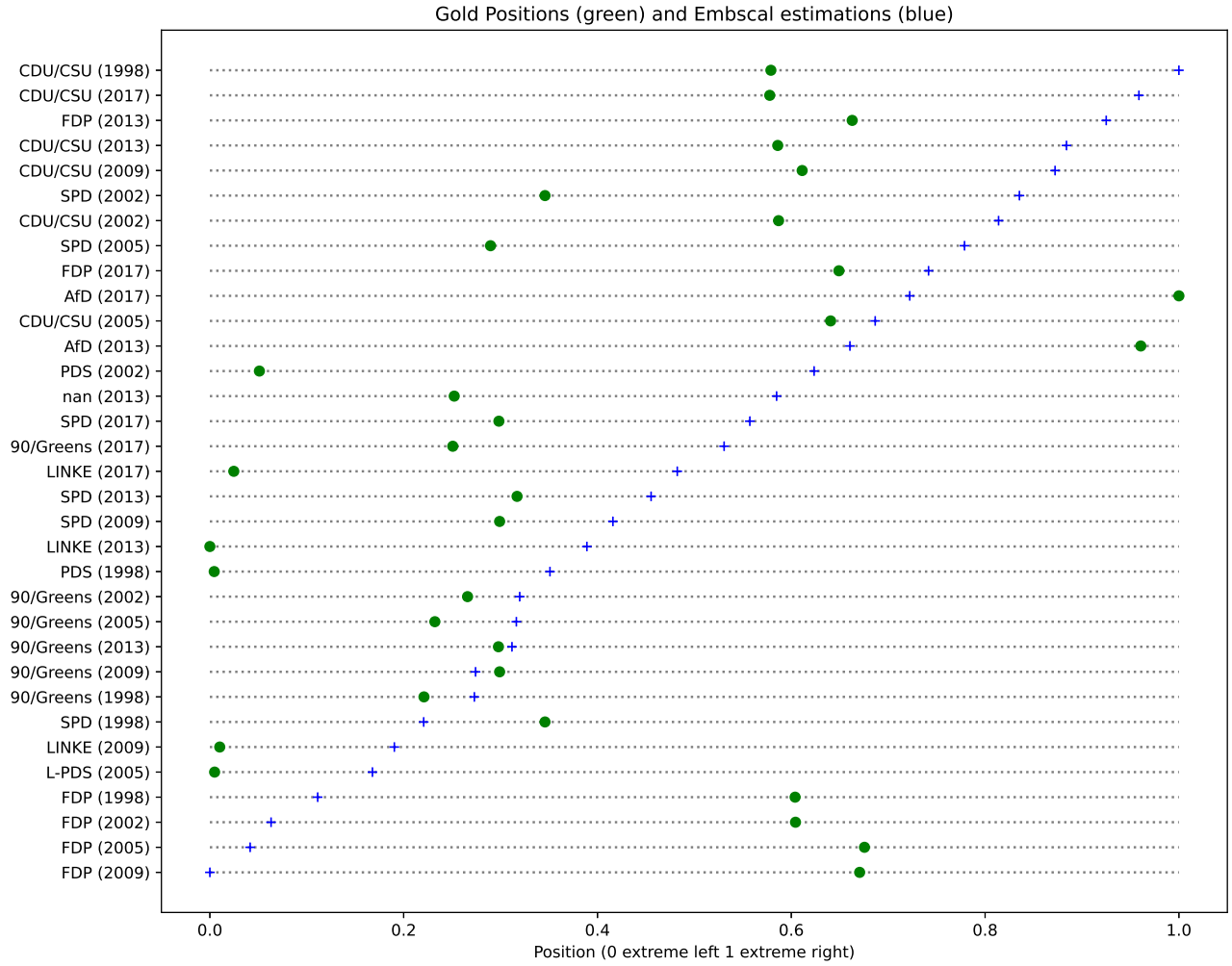

Figure A.4: Embscal estimations compared to the gold positions of the manifestos from different parties. In parantheses is the election year to which the manifesto was published. both positions are min-max scaled to be between 0 and 1.

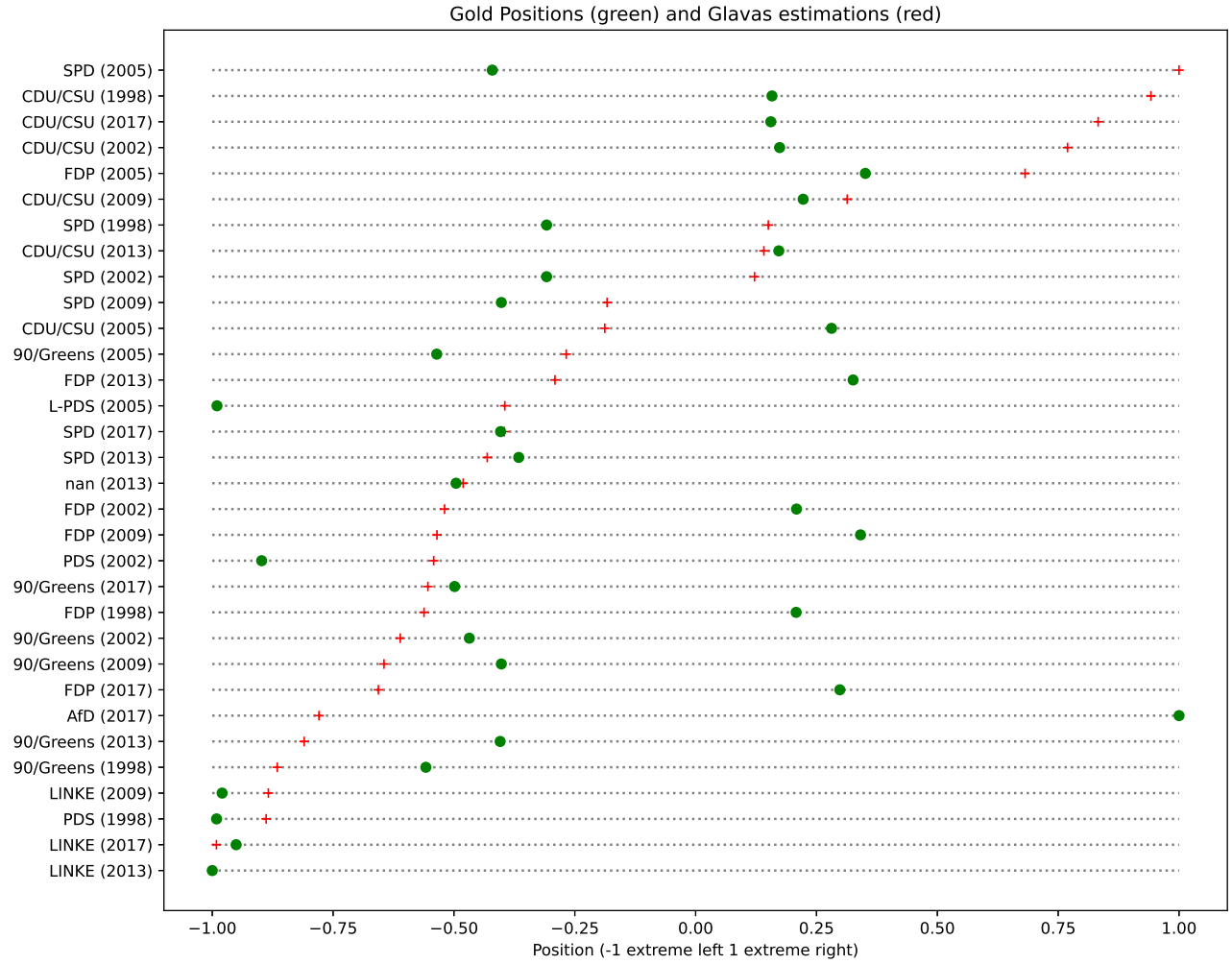

Figure A.5: Glavas method estimation with the e5 embeddings and alignment Similarity. In parantheses is the election year to which the manifesto was published. both positions are min-max scaled to be between -1 and 1.

Top 10 Left Words	β	**Top 10 Right Words**	β
gesundheitsversicherung	5.9483	bewußtsein	-6.1510
linker	5.1154	bundesgesetze	-5.1436
erschöpfung	4.9606	bundesbeteiligungen	-4.9583
mindestsicherung	4.6155	verläßlicher	-4.9037
feierabend	4.5108	steinkohlesubventionen	-4.8807
sozialticket	4.4903	beitragsstabilität	-4.7849
kuba	4.4903	staatstätigkeit	-4.7363
antifaschistische	4.4903	zusammengefaßt	-4.6802
zukunftsprogramm	4.4260	verläßlichkeit	-4.6446
gemeindewirtschaftsteuer	4.2330	staatswirtschaft	-4.5385

Table A.2: The Top 10 Words Wordfish identified as placing a manifesto on one end of the political spectrum

Method	**failed**	r_P	r_S	τ	**PA**
Random	0	0.4831	0.4781	0.3876	68.41%
Mistral7b (Wiki 4096)	2	0.5489	0.5318	0.4662	64.92%
Llama3 8b (Wiki 4096)	7	0.7799	0.7186	0.6314	75.00%
Gemma2 9b (Wiki 4096)	4	0.6454	0.6622	0.5992	66.75%
Llama3.1 8b (LLM 4096)	6	**0.8208**	0.7621	0.6686	63.89%
Llama3.1 8b (LLM 100k)	4	0.6478	0.6779	0.6558	**80.00%**
Gemma2 9b (LLM 4096)	8	0.7372	0.677	0.6266	65.00%
Mistral 7b (LLM 7k)	8	0.6850	0.7130	0.6447	71.11%
Mistral 7b (LLM 4096)	**1**	0.8051	**0.7774**	**0.7210**	71.11%
Glavas (ALIGN) (multilingual-e5-large-instruct)	0	0.5464	0.4244	0.4137	69.44%
Glavas (ALIGN) (e5-mistral-7b-instruct)	0	0.3963	0.2294	0.2731	62.78%
Glavas (AVG) (e5-mistral-7b-instruct)	0	0.6129	0.5345	0.4624	72.06%
Glavas (AVG) (multilingual-e5-large-instruct)	0	0.5384	0.4798	0.3894	68.41%
Embscal-umap-default	0	0.5522	0.5554	0.4516	72.35%
Embscal-tsne-default	0	0.5306	0.5133	0.3643	67.99%
Wordfish-unlemmatized-R	0	0.8498	0.8192	0.6992	83.81%
Wordfish-lemmatized-R	0	0.6979	0.6537	0.5293	75.32%

Table A.3: The scaling performance of the different Models, when only manifestos from the same year are scaled and evaluated together (YEARLY). r_P and r_S being Pearson and Spearman correlation coefficent. τ is the Kendall-Tau and PA stands for pairwise accuracy

```
Du bist ein LLM, das Bundestagsreden einem
   einzigen passenden Thema aus einer
   vorgegebenen Liste zuordnet. Deine Aufgabe ist
   es, das Thema auszuwählen, das den Inhalt und
   die Kernaussagen der gesamten Rede am besten
   widerspiegelt. Beachte dabei folgende Regeln:

Themenauswahl:
    Wähle nur ein einziges Thema aus der Liste
       aus.
    Das gewählte Thema muss exakt aus der
       vorgegebenen Liste stammen.
    Du darfst keine neuen Themen erfinden oder
       hinzufügen.

Überprüfung:
    Überprüfe sorgfältig, ob dein gewähltes Thema
       wirklich in der Liste enthalten ist.

Antwortformat:

    Halte dich strikt an folgendes Schema:

    Thema:(Hier das gewählte Thema aus der Liste
       einfügen)

    Begründung:
    (Hier erläuterst du detailliert, warum dieses
       Thema am besten zur Rede passt.
       Beschreibe, welche zentralen Aussagen der
       Rede deine Wahl stützen.)

Hier ist die Rede:
{Document}

Hier ist die Liste der Themen:
{tree}

Dein Antwort:
```

Listing A.1: The topic assignment prompt. The speech to classify gets inserted at {Document} and the List of topics at {tree}

```
Du bist ein LLM das einer Gruppe von
   Bundestagsreden mit einem Thema beschreiben
   soll. Die Gruppe ist durch eine  eine Liste
   von Schlüsselwörtern beschrieben.
Das Thema wird von den folgenden Schlüsselwörtern
   beschrieben:
{KEYWORDS}

Bedenke das es viele Perspektiven auf die Themen
   gibt daher solltest du möglichst neutrale
   Bezeichnungen finden.
Du darfst nur exakt ein Thema finden.
Das Thema muss kurz und prägnant sein.
Das Thema muss zu den Schlüsselwörtern passen
Das Thema sollte so allgemein sein dass es auf
   Reden von allen politischen Richtungen passt.

Deine Antwort muss die Folgende Struktur haben:
Thema: hier das Thema
Begründung: hier die Begüundung
Bitte trenne Antwort und Begründung klar durch
   einen Absatz.
```

Listing A.2: The prompt that asks the the LLM to find a fitting title for a topic based on the BERTopic representation. The words found by BERTopic as representation get inserted at{KEYWORDS}

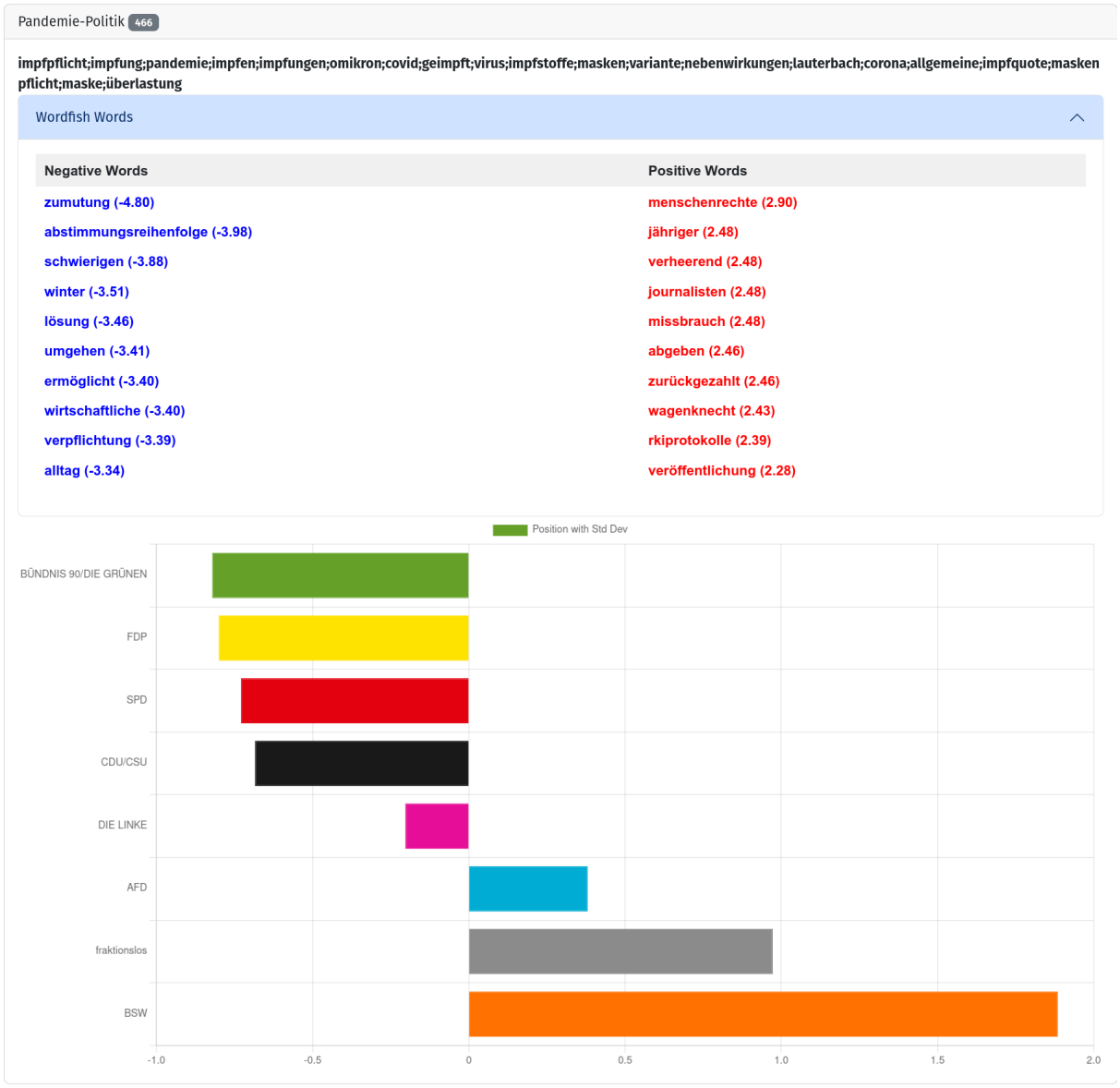

Figure A.6: The parties' position analysis for the topic of covid. Above the bar chart are the relevant Words. The position estimations correlate well with the actual positions of the parties

```
Ich habe ein Thema das unter anderem folgende
   Dokumente enthält: \n {Dokumente}.
Das Thema wird von den folgenden Schlüsselwörtern
   beschrieben: {KEYWORDS}

Welches Thema aus dieser Liste an Themen ist am
   passensten zu den Schlüsselwörtern:
{Liste}

Du darfst nur exakt Themen aus der Liste
   auswählen. Unter keinen Umständen eigene
   Themen erfinden.
Du Darfst nur genau ein Thema angeben.
Deine Antwort muss die Folgende Struktur haben:
Thema: hier das aus der Liste ausgewählte Thema
Begründung: hier die Begüundung
Bitte trenne Antwort und Begründung klar durch
   einen Absatz.
```

Listing A.3: The topic representation prompt prompt. The most representative speeches for that topic get inserted at {Dokumente} and the representation found by LDA or BERTopic gets inserted at {KEYWORDS} and the List of topics at {Liste}

```
Ich habe das folgendes Thema:
{TOPIC}
Ich habe Wordfish auf allen Bundestagsreden zu
   diesem Thema angewandt. Deine Aufgabe ist es
   Sinvolle Bezeichnungen für die Beiden von
   Wordfish extrahierten extrempositionen
   (negativ und positiv) zu finden. Dazu erhälst
   du die Wärter die Wordfish als besonders
   relevant für die jeweiligen extreme gefunden
   hat:
negative Wörter:
{negative_words}
positive Wörter:
{positive_words}
Außerdem erhälst du die Reden die von Wordfish
   als extremsten eingestuft wurden.
negative Extrem Reden:
{negative_speeches}

negative Extrem Reden:
{negative_speeches}

Bitte Finde Aussagekräftige kurze bezeichnungen
   für die negative position und die Positive
   Position.
Bedenke dass es darum geht ein spektrum zu
   beschreiben bei dem die vorliegenden Reden nur
   die Extrempositionen sind.
Bitte antworte nach folgendem Schema:
Negatives Extrem Bezeichnung: "hier deine
   Bezeichnung",
Begründung für negative Bezeichnung: "hier deine
   kurze Begründung",

Positive Extrem Bezeichnung: "hier deine
   Bezeichnung",
Begründung für positive Bezeichnung: "hier deine
   kurze Begründung",

Bitte sei präzise und objektiv in deiner Antwort.
Die Bezeichnungen müssen einen Bezug zum Theme:
   '{TOPIC}' haben
```

Listing A.4: The prompt given to GPT4.0 to find descriptions for the Extreme positions. The topic Title is inserted at {TOPIC }. The relevant Words found by Wordfish are inserted at {negative_words} and {positive_words}. The representative speeches are inserted {negative_speeches} and {positive_speeches}

```
Negatives Extrem Bezeichnung:
   "Integrationsförderer",
Begründung für negative Bezeichnung: Die Reden
   und Wörter im negativen Extrem konzentrieren
   sich auf die Förderung von Integration,
   Bildung und Wissenschaft für Flüchtlinge,
   insbesondere durch Unterstützung von
   wissenschaftlichen Kooperationen,
   Bildungssystemen und akademischen
   Einrichtungen. Sie legen Wert auf die
   Integration von Migranten als Teil einer
   positiven Anpassung an Migrationsbewegungen.

Positive Extrem Bezeichnung:
   "Migrationseinschränker",
Begründung für positive Bezeichnung: Die Reden
   und Wörter im positiven Extrem betonen die
   Notwendigkeit von restriktiveren Maßnahmen
   gegen Migration. Hervorgehoben werden
   Belastungen für das Sozialsystem durch
   unkontrollierte Migration und die Forderung
   nach Grenzschutz, um die Zuwanderung zu
   reduzieren und potentielle negative Effekte
   auf die Gesellschaft zu begrenzen.
```

Listing A.5: The answer of GPT4.0 if prompted to find extreme position labels for the topic migration

```
Negatives Extrem Bezeichnung: "Herausforderungen
   und Missstände im Wohnungsmarkt",
Begründung für negative Bezeichnung: Die negative
   Position konzentriert sich auf Probleme und
   Missstände im Wohnungsmarkt, wie die Thematik
   um Schrottimmobilien, problematische
   Zwangsversteigerungen, Mietwucher und die
   Ineffizienz der Mietpreisbremse. Diese
   Begriffe und Themen spiegeln die Sorgen und
   Herausforderungen wider, denen Mieter und
   Anwohner gegenüberstehen.

Positive Extrem Bezeichnung: "Strategien zur
   nachhaltigen Wohnraumgestaltung",
Begründung für positive Bezeichnung: Die positive
   Position ist durch Begriffe geprägt, die sich
   mit ökologischen und strategischen Ansätzen
   zur Wohnraumentwicklung beschäftigen, wie
   Klimaanpassung, Flächenentwicklung und die
   Rolle der BImA. Diese Themen reflektieren eine
   langfristige und umfassende strategische
   Vision für die nachhaltige und
   zukunftsorientierte Gestaltung des Wohnraums.
```

Listing A.6: The answer of GPT4.0 if prompted to find extreme position labels for the topic housing supply

```
Negatives Extrem Bezeichnung:
   "Ökologisch-Integratives Modell",
Begründung für negative Bezeichnung: Diese
   Position unterstreicht die Bedeutung
   nachhaltiger Praktiken und den Erhalt ö
   kologischer Systeme, wobei sie den Fokus auf
   gesunde Ernährung, Gemeinschaftsverpflegung
   und regionale Initiativen legt.

Positive Extrem Bezeichnung:
   "Marktwirtschaftlich-Protektives Modell",
Begründung für positive Bezeichnung: Diese
   Position kritisiert staatliche Eingriffe wie
   Steuererhöhungen und betont den Schutz
   traditioneller landwirtschaftlicher Strukturen
   gegen solche Maßnahmen, was oft mit dem Schutz
   vor wirtschaftlicher Belastung und globalem
   Wettbewerb in Verbindung gebracht wird.
```

Listing A.7: he answer of GPT4.0 if prompted to find extreme position labels for the topic agriculture

```
Bitte analysiere das soeben genannte Wahlprogramm
   und ordne es ideologisch auf einer Skala von 1
   bis 10 ein, wobei:

    1 für "sehr links" steht
    10 für "sehr rechts" steht
Links und Rechts ist Politisch gemeint und wie
   folgt definiert:

Im Allgemeinen zeichnet sich der linke Flügel
   durch eine Betonung von Ideen wie Freiheit,
   Gleichheit, Brüderlichkeit, Rechten,
   Fortschritt, Reform und Internationalismus
   aus, während der rechte Flügel durch eine
   Betonung von Begriffen wie Autorität,
   Hierarchie, Ordnung, Pflicht, Tradition,
   Reaktion und Nationalismus gekennzeichnet ist.

Deine Bewertung soll sowohl ökonomische (z. B.
   Verteilungspolitik, Marktregulierung) als auch
   gesellschaftliche (z. B. Werte, Migration,
   Sicherheit) Aspekte berücksichtigen.

Stelle sicher, dass die Analyse länger und
   inhaltsreicher ist. Füge Beispiele aus dem
   Wahlprogramm ein (falls möglich) und begründe
   die Bewertung ausführlich. Der Output muss
   klar im folgenden JSON-Format erfolgen:

{
  "score": <integer von 1 bis 10>,
  "comments": [...],
}

Beispiel für die Ausgabe:

{
  "score": 7,
  "comments": [...],
}
Trenne Score und Begründung Klar wie im
   antwortformat außerhalb der JSON antwort
   darfst du nichts hinzufügen.
Dein JSON Output hier:
```

Listing A.8: The prompt for the stance detection with LLMs with a defintion provided by GPT4.0

```
Bitte analysiere das soeben genannte Wahlprogramm
   und ordne es ideologisch auf einer Skala von 1
   bis 10 ein, wobei:

1 für "sehr links" steht, was tendenziell
   progressive, egalitäre, sozialistische oder
   kollektivistische Ansätze betont (z. B.
   staatliche Umverteilung, soziale
   Gerechtigkeit, ökologische Nachhaltigkeit,
   starke Regulierung von Unternehmen).
10 für "sehr rechts" steht, was tendenziell
   konservative, marktliberale, nationalistische
   oder hierarchische Ansätze betont (z. B.
   privates Unternehmertum, reduzierte staatliche
   Eingriffe, Tradition, Betonung nationaler
   Identität).

10 für "sehr rechts" steht, was tendenziell
   konservative, marktliberale, nationalistische
   oder hierarchische Ansätze betont (z. B.
   privates Unternehmertum, reduzierte staatliche
   Eingriffe, Tradition, Betonung nationaler
   Identität).
```

Listing A.9: The definition provided by GPT4.0 for left and right

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Bachelorstudiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe. Sofern im Zuge der Erstellung der vorliegenden Abschlussarbeit generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten Wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Unterschrift: *Ort, Datum:*

Erklärung zur Veröffentlichung

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik zu.

Unterschrift: *Ort, Datum:*