Proceedings of the GermEval 2023 Shared Task on Speaker Attribution in Newswire and Parliamentary Debates (SpkAtt-2023).
September 19, 2023, Ingolstadt, Germany.

Politics, BERTed: Automatic Attribution of Speech Events in German Parliamentary Debates

Anton Ehrmanntraut
Julius-Maximilans-Universität Würzburg
anton.ehrmanntraut@uni-wuerzburg.de

Abstract

This paper documents and analyzes a submission to the Shared Task on Speaker Attribution hosted at KONVENS 2023 (Rehbein et al., 2023). One task was the automatic identification of speech events in German parliamentary debates, i.e., where speech, thought or writing is referenced by speakers of parliament. The system approaches this with a token and sequence classification setup and offers a BERT-based solution to this task. According to the results, the proposed system performs surprisingly well despite its simple architecture. Further experiments indicate that even with a smaller variant of BERT, the system performs nearly equally well, whereas a domain adaptation of BERT on parliamentary speeches offered close to zero improvement.

1 Introduction

This paper presents a participating system at the *KONVENS 2023 Shared Task on Speaker Attribution (SpkAtt-2023)*, particularly participating in the task 1 on German parliamentary debates. The goal of the shared task is the automatic identification of speech events in political debates (whereas, for task 2, in news articles) and attributing them to their respective speakers, essentially identifying who says what to whom in the parliamentary debates (Rehbein et al., 2023). This is motivated by the fact that the automatic identification of such information is a prerequisite for an extensive semantic analysis of unstructured texts. For instance, the information automatically inferred from parliamentary speeches could be used for political discourse studies of parliamentary debates, or political communication.[1]

A *speech event* refers to a reference to speech, writing or thought by a member of parliament during one of their plenary speeches. Each such speech event consists of several word spans: first, a nonempty span of *cue words* that trigger this speech event (usually a verb), and second, several *role spans* associated with this speech event, i.e., *Source, Addressee, Message, Topic, Medium, Evidence*, or *Particles*, any of which can be empty, and all may pairwise overlap.[2] See Figure 1 for some examples. In this sense, a speech event does not have to be attributed to the actual person delivering the speech in parliament: the person may, for example, also state the thoughts of another entity, such as depicted in Figure 1(b).

The system presented in this paper approaches automatic speaker attribution through multiple fine-tuned BERT Transformer models (Devlin et al., 2019), designed to handle cue detection, cue linkage, and role detections. The system is specifically designed to be a minimal BERT-based baseline; all involved NLP tasks are essentially simple token classifications resp. sequence classifications. The model is similar to a semantic role labeling model by Shi and Lin (2019); in both models, entire sentences were encoded to leverage the contextual information from all tokens in the sequence at the same time.

The system was trained on the GePaDe dataset[3] for speaker attribution in German parliamentary debates, which has been specifically created for the SpkAtt-2023 task. It consists of 265 speeches, mostly from the 19th legislative term of the German Bundestag. For the shared task evaluations, the task organizers tested the submitted systems on (blind) test data. According to the official scorer, the presented system achieved a SpkAtt-F1 score of 0.83 on full inference (subtask 1a), and a SpkAtt-F1 score of 0.92 on a simplified task where gold cue

[1] See also the GePaDe datasheet: https://github.com/umanlp/SpkAtt-2023/blob/master/doc/SpkAtt-Debates-Datasheet.pdf.

[2] See also the precise annotation guidelines: https://github.com/umanlp/SpkAtt-2023/blob/master/doc/Guidelines_SpeakerAttribution_in_Parliamentary_Debates-SpkAtt-2023_Task1.pdf.

[3] https://github.com/umanlp/SpkAtt-2023

Figure 1: Example instances for the speaker attribution task. Note how the cue span can cover multiple tokens in a non-contiguous way (b). Note how, in the same speech event, words can be assigned to multiple role spans (c; from the GePaDe training set `ID197411900`). Also note how two annotations may overlap, how annotations may span multiple sentences (d and e), and how multiple annotations can be present even in the same sentence (e and f).

words are already given (subtask 1b). The entire system is made available.[4]

2 Related Work

Speaker attribution has many parallels to semantic role labeling. Similar to speaker attribution, semantic role labeling refers to the task of identifying the predicate of a clause, establishing "what" took place (typically a verb) and the associated arguments that specify the "who," the "what," "where," etc. Like the speaker attribution system presented here, semantic role labeling is usually divided into four steps: predicate identification and disambiguation, and argument identification and classification (Conia and Navigli, 2022). Current state-of-the-art semantic role labeling models build upon large pretrained language models such as BERT. In particular, the current best-performing model operating on German appears to be the multilingual one developed by Conia et al. (2021; see also Conia and Navigli, 2020).

Nevertheless, we have indications that much simpler models for semantic role labeling perform quite close to the state of the art. For instance, Conia and Navigli (2020) report that the monolingual BERT baseline model provided by Shi and Lin (2019) performs nearly equally as good as their more complex (multilingual) model on English. Essentially, the model by Shi and Lin performs argument identification by taking BERT's output representation and feeding it through a BiLSTM layer to predict BIO-encoded predicate labels.

However, they only fine-tune the BiLSTM layer; the attention weights of BERT remain fixed. Current research on Named Entity Recognition (Schweter and Akbik, 2021) and—closer to the speaker attribution task—recognition of speech, thought, and writing representation (Ehrmanntraut et al., 2023) suggests that rather than adding a BiLSTM layer, fine-tuning the Transformer's attention weights allows to predict the respective labels from the token encoding in the final Transformer layer alone. This Transformer-Linear variant corresponds to the now usual "BERT for token classification" setup and appears to be competitive, and often even outperforming model variants with BiLSTM decoders. My system directly follows this approach.

3 Base Model and Domain Adaptation

My system is based on the BERT Transformer model $\text{GBERT}_{\text{Large}}$ (i.e., `deepset/gbert-large`, Chan et al., 2020). Following Gururangan et al. (2020) and Konle and Jannidis (2020), I performed a domain adaptation of the model by continuing pretraining on a second, separate corpus of speeches. The corpus extends the SpkAtt training speeches with additional speeches held in the German Bundestag during the 9th–20th legislative period, from 1980 until April 2023 (757 MB). This results in the BERT model GePaBERT. The speeches were automatically prepared from the publicly available plenary protocols[5], using the extraction pipeline Open Discourse[6] (cf. Richter et al., 2023). Speeches that are present in the development or test split of the SpkAtt task were excluded, so that the predictive accuracy measured on the held-out development/test split actually reflects the accuracy on data the sys-

[4] https://github.com/aehrm/spkatt_gepade

[5] https://www.bundestag.de/services/opendata

[6] https://opendiscourse.de;
https://github.com/open-discourse/open-discourse

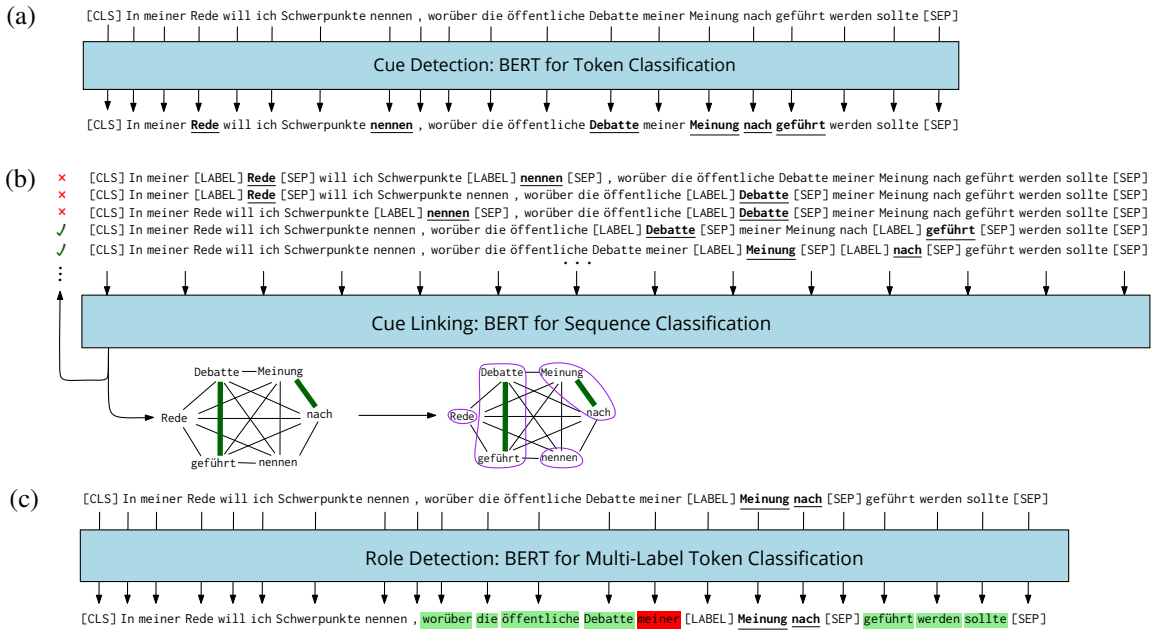

Figure 2: Overview of the system architecture. (a) The first component performs a token classification to detect cue words. (b) The second component performs a sequence classification to predict potential cue links on all pairs of cue words through contextualized cue-aware input sequences. (Not all such sequences are shown.) On the graph induced by the cue words resp. positive (green) links, the system picks the connected components (circled purple) as cue spans. (c) The third component performs a multi-label token classification to detect role words corresponding to the respective highlighted cue span.

tem has never seen at all, e.g., speeches held after April 2023.

Training was done on 5 epochs, with a batch size of 8, and a learning rate of 2×10^{-5}, linearly decreasing to zero. (Training took approximately 140 GPU hours on two GTX 1080 TI GPUs, each with a device batch size of 2, and 2 gradient accumulation steps.) The final model GePaBERT is made available on the Huggingface hub.[7]

4 System Overview

My system splits the task into three components: (a) Detection of cue word, i.e. word that are covered by cue spans. (b) Joining individual cue word through the detection of cue links, in order to form cue spans. (c) For each cue span, given that specific cue span, infer the associated role spans. Figure 2 gives a sketch of the system. All three components are implemented by fine-tuning the above domain-adapted BERT model GePaBERT, respectively, employed in a token classification or sequence classification setup.

Instead of fully fine-tuning BERT models, the system builds upon LoRA adapters (Hu et al., 2021): rather than training all Transformer weights, the pre-trained weights are frozen, but trainable rank decomposition matrices are injected into each attention layer of the Transformer architecture. This reduces the number of trainable parameters and accelerates fine-tuning. To this end, the system is implemented through the PEFT library provided by the Huggingface API[8] (Mangrulkar et al., 2022; Wolf et al., 2020).

4.1 Cue Detection

The detection of cue words is achieved using a token classification by the first BERT model, fine-tuned for this task. Following standard practice, the model performs a token-level binary logistic regression, using BERT's output representation of the respective first wordpiece token of that particular word. Thus, the models differentiates between non-cue words and cue words. In this component and all the following, for each the regression weights and the respective Transformer's attention weights (through LoRA) are trained to minimize the binary cross entropy loss of the token classification against gold labels. Training was done over 30 epochs with a batch size of 4 and a learning rate of 5×10^{-5}. (In total, fine-tuning all three components took ap-

[7] https://huggingface.co/aehrm/gepabert

[8] https://github.com/huggingface/peft

proximately 6 GPU hours on a single GTX 1080 TI GPU.)

The model performs this token classification for each sentence, but adds additional context by prepending the five preceding sentences, and appending the five following sentences to the sequence, both during training and inference. After inference, we obtain a set of predicted individual cue words.

4.2 Cue Linking

The second component joins individual cue words into cue spans. This is necessary since, even within the same sentence, there may be multiple different cue spans, each with their own associated role spans. (See, e.g., Figure 1(e) vs. (f).) To this end, the component predicts whether two cue words belong to the same span. Given two cue words, we first derive a cue-aware input sequence that highlights the two cue words. Then, we let the second BERT model perform a sequence classification on the input sequence, predicting whether the two highlighted cue words belong to the same span or to different spans. During inference, these link predictions are used to calculate a partition of cue words into spans.

In order to encode the two focused cue words into a cue-aware manner, a new special token `[LABEL]` was introduced, and the input sequence is designed as "`[CLS]` left context `[LABEL]` cue no. 1 `[SEP]` center context `[LABEL]` cue no. 2 `[SEP]` right context `[SEP]`". As usual, the sequence prediction is calculated using a binary logistic regression on BERT's output representation of the initial `[CLS]` token.

This classification is performed on all pairs of cue words that appear in the same sentence. (In fact, no cue span appears to span over multiple sentences.) The model is trained on all gold cue words, predicting whether two focused cue words are indeed contained in the same (gold) cue span. Again, the five preceding/following sentences were added to the left/right context.

During inference, the model takes the cue words predicted by the previous component. To now derive the actual partition into cue spans, set up a graph structure with every (predicted) cue word as vertex, and adding edges between two vertices if the classifier predicted a link between the two respective cue words. Finally, the model enumerates the connected components of that graph as prediction for the cue spans. While an enumeration of maximal cliques would also be an option—especially since under gold predictions, the connected components are always cliques—the component relaxes this condition and focuses only on connected components. In fact, no noticeable difference in performance between these two approaches could be observed.

4.3 Role Detection

The last component predicts the role spans, *given* a specific cue span, in the context surrounding the cue span. Like the first component, this component fine-tunes the third BERT model to perform a multi-label token classification, which, for each token, predicts to which role span(s) the respective token belongs. Note that a multi-label classification is needed since, even in the same speech event, a word may belong to *multiple* different role spans, even when associated with the same cue span. (Cf. Figure 1(c), which appears verbatim in the official GePaDe training dataset.)

This multi-label classification is modelled as seven independent binary classifiers (one for each role label *Source*, *Message*, *Topic*, …, i.e., binary relevance method). Again following standard practice, similar to the cue detection, each one of the classifiers is implemented as independent token-level binary logistic regression on (the same) output representation from BERT.

As input sequence, the model takes the sentence that contains the cue span, plus the five preceding and following sentences: for one, since role spans could also cover tokens from preceding/following sentences (cf. Figure 1(d)); for another, to give BERT more context to, e.g., disambiguate what the demonstrative pronoun *das* in Figure 1(b) actually refers to. The model encodes the sequence in a cue-aware manner similar to the previous component: again, the special token `[LABEL]` highlight contiguous tokens from the cue span. For instance, the speech event depicted in Figure 1(b) would be encoded as `[CLS] Frau Merkel , laut Medien [LABEL] nahm [SEP] die Bundesregierung das aber nicht [LABEL] zur Kenntnis [SEP] . [SEP]`.

This component has been trained on the gold annotation objects (i.e., given gold cue span, predict the gold role spans). During inference, the component takes the cue spans predicted by the previous component, and for each cue span, predicts the associated role spans, and finally returns complete

annotation objects by combining the cue spans with the respective predicted role spans.

5 Results and Error Analysis

5.1 Metric

Since no code for the official SpkAtt scorer is available, we will, in the further course of the paper, instead resort to the following *matching-based precision/recall* as a guiding metric, which should be approximately in line with the informal descriptions given by the task organizers.

Consider one gold annotation A and one predicted annotation $\hat{A}$. For each of the eight shared task classes (*Cue*, *Source*, *Message*, etc.), calculate the recall between the gold span and the predicted span, each time on the token level.[9] Then, form the micro-averaged recall over all classes to get the *annotation recall* $R(A, \hat{A})$ between gold and predicted annotation.

Now, to calculate recall between *sets* of gold annotations and predictions, set up a complete bipartite graph between gold annotations and predictions. Weight each edge between gold A and predicted $\hat{A}$ according to $R(A, \hat{A})$. Then, determine a maximum-weight matching in that bipartite graph. The *matching-based recall* is the average of $R(A_i, \hat{A}_{i'})$, taken over all gold annotations A_i, where $\hat{A}_{i'}$ is the matched mate of A_i. (If A_i has no mate, then it contributes recall 0 to the average.)

Precision is computed in a symmetric fashion. Calculate micro-averaged annotation precision $P(A, \hat{A})$, and then calculate the maximum-weight matching with respect to a bipartite graph weighted by $P(A, \hat{A})$. The *matching-based precision* is the average over $P(A_{i'}, \hat{A}_i)$ taken over all predicted annotations $\hat{A}_i$, where $A_{i'}$ is the matched mate of $\hat{A}_i$. (Again, if $\hat{A}_i$ has not mate, it contributes precision 0 to the average.)

Now the matching-based F1 score Match-F1 is the harmonic mean between matching-based precision and recall. Note that the maximum-weight matchings calculated for precision resp. recall may not be identical.

5.2 Quantitative Results

The organizers designed the task with two subsettings: In the full task (1a), predict cue spans together with corresponding roles. In the role labeling task (1b), the gold cue spans are given, and the task consists in predicting only the corresponding roles.

	Full Task (1a)			**Gold cues given (1b)**		
	Match-Prec.	Rec.	F1	Prec.	Rec.	F1
Dev Set	84.3	85.2	84.8	92.7	92.1	92.4
only cues	90.8	92.2	91.5	—	—	—
only roles	81.0	83.1	82.0	87.9	89.3	88.6
	SpkAtt-Prec.	Rec.	F1	Prec.	Rec.	F1
Test Set	78.9	87.3	82.8	92.1	91.3	91.7
only cues	89.7	88.9	89.3	—	—	—
only roles	77.7	87.1	82.1	91.1	90.2	90.7

Table 1: Results of the system on the development set (evaluated using Match-Precision/Recall/F1) and on the test set (evaluated by the task organizers, using their SpkAtt-Precision/Recall/F1). "Only cues" resp. "only roles" refers to the metric variant where only cue spans resp. role spans are considered in the calculation. All scores are given in percentage points.

The proposed system was evaluated on two datasets: one, the provided development split of the GePaDe dataset. Second, on a blind test split, for which the gold annotations were only available to the shared task organizers. On both datasets, the system was evaluated with respect to both subtasks 1a and 1b. However, the metrics employed differ between the datasets: for the test set, the gold annotations are not publicly available, thus only the metrics returned by the task organizers are reported, denoted by SpkAtt-Precision/Recall/F1, who ran their closed-source official scorer on the submitted predictions.[10] In the development split, I used only the matching-based precision/recall as outlined above in Sec. 5.1, denoted with Match-Precision/Recall/F1.

Table 1 presents the results on the two datasets and the two task settings. Broadly, the results suggest that, even in this relatively simple setup, this BERT-based baseline already gives surprisingly steady performance. As we expect, this even in-

[9] I.e., when s is the gold span and s' is the predicted span of a particular class, then $|s \cap s'|$ is the number of true positives, $|s \setminus s'|$ is the number of false negatives, and $|s' \setminus s|$ is the number of false positives.

[10] The two respective predictions were submitted to CodaLab on July 30, 23:00 (No. 16) for task 1a and on August 2, 12:08 (No. 19) for task 1b. For task 1a, I thus report the performance of the second-last submission, not the last submission for task 1a (No. 17) which the task organizers intended to treat as the final official submission for task 1a. This final submission for task 1a differs to the one reported here only in the cue linking algorithm; the respective performances are nearly identical. (82.73 vs. 82.84 SpkAtt-F1 points for the system reported here.)

	Prediction with gold cues given (1b)			
Role Class	Match-Prec.	Rec.	F1	# train instances
Source	93.3	96.4	94.8	3337
Message	88.6	91.3	89.9	3242
Topic	70.8	83.1	76.4	871
Addressee	75.9	91.3	82.9	495
Particle	88.4	90.5	89.4	359
Medium	65.7	78.3	71.5	228
Evidence	77.8	69.9	73.6	80

Table 2: Breakdown on the system's performance on the development set in the role labeling task, when gold cues are given (1b), where metrics are calculated for each individual role class. The last column refers to the number of role spans per class present in the training split. All scores are given in percentage points.

creases in the second subtask (1b), where the gold cues triggering the speech events are given.

Table 2 shows the performance for the role labeling task (1b) on the development set, broken down for each of the seven role classes. We can observe a clear trend that classes occurring less frequently in the training set are recognized less accurate. Additionally, a more detailed quantitative analysis (not shown here) indicates that the system slightly struggles to differentiate between *Topic* vs. *Message*, and *Medium* vs. *Message*.

To further assess the impact of domain adaptation of the chosen base BERT model and the variability introduced by the random fine-tuning, I repeated the fine-tuning five times on GePaBERT, but also on $\text{GBERT}_{\text{Large}}$ (i.e., GePaBERT before domain adaptation), and $\text{GBERT}_{\text{Base}}$ (i.e., the smaller variant `deepset/gbert-base` with fewer layers). Note that this was only conducted after the shared task's system submission deadline. Table 3 reports the measured accuracies, given in empirical mean and standard deviation.

As we expect, we clearly observe a jump in performance between the "base-size" and "large-size" variant of BERT. However, the domain adaptation of $\text{GBERT}_{\text{Large}}$ to GePaBERT, as outlined in Section 3, appears to have only minimal or no effect at all. I do not have a good explanation for this behavior. For one, maybe more data is necessary for an effective domain adaptation; for another, perhaps further hyperparameter studies for the domain adaptation are necessary to find the optimal pre-training procedure. Along this, pre-training itself should also be extended beyond the current five epochs, something for which there was insufficient time during the development of the system. Or, arguing in the other direction, the observed performances by both the GePaBERT and $\text{GBERT}_{\text{Large}}$ might suggest that both models already hit the same performance ceiling, which might be much harder to break through.

While these results contradict the findings of Konle and Jannidis (2020)—who were able to achieve substantial improvements using domain adaptation—it should be noted that they also included the test set of the corresponding downstream task during the pre-training of the base language model (though not during the fine-tuning). As explained in Section 3, this was not done for the system at hand, in order to measure accuracy against future data that the model has never seen. Yet, as Konle and Jannidis hypothesize, precisely this pre-training on the (unlabeled) test data may allow the language model to build a better representation of the test data, helping in solving the downstream task. Nevertheless, such an increase in accuracy comes with the disadvantage that, when the system is applied on new unlabeled data, the entire base language model may possibly need to be pre-trained again on this new data to maintain the same performance. In total, further research towards domain adaptation (especially in Computational Humanities resp. Computational Social Sciences) is needed.

5.3 Qualitative Error Analysis

Next to the quantitative analysis of the system's performance, I also performed a manual error analysis of the system's predictions on the development set. Concerning the cues, it appears that the system is particularly struggling with recognizing nominal triggers, e.g., "*Als nächster Redner hat das Wort [...],*" "*Wo waren Sie bei den Koalitionsverhandlungen?,*" "*Die richtige Antwort bei Betrug, [...]*" have not been predicted as cues, whereas the system erroneously predicts, e.g, "*Die Ziele des Gesetzentwurfs sind nicht einmal falsch, [...],*" "*An dieser Einsicht hat sich [...] nichts verändert,*" etc. Furthermore, in many of the false-positive cases, the presence of speech, thought, or writing representation, is ambiguous, e.g., in "*Die Mehrzahl der Handwerksbetriebe beurteilt [...] die wirtschaftliche Lage als sehr gut"* the verb is predicted as cue, but not annotated as such.

Concerning the role prediction, I am focusing on the results for the task setting where gold cues are given (1b). The manual analysis confirms the observation already outlined above, that the system is

Model	Full Task (1a) Match-F1	Full Task (1a) (on cues only)	Full Task (1a) (on roles only)	Gold cues given (1b) Match-F1	Gold cues given (1b) (on roles only)
$\text{GBERT}_{\text{Base}}$	80.09 ± 1.12	90.06 ± 0.54	75.23 ± 2.60	88.66 ± 1.44	82.60 ± 3.09
$\text{GBERT}_{\text{Large}}$	$\mathbf{84.16 \pm 0.98}$	90.84 ± 0.78	$\mathbf{81.76 \pm 1.02}$	$\mathbf{92.07 \pm 0.60}$	$\mathbf{88.07 \pm 0.90}$
GePaBERT	84.12 ± 0.74	$\mathbf{91.36 \pm 0.44}$	81.24 ± 1.07	91.55 ± 0.75	87.18 ± 1.13

Table 3: F1-Scores on five fine-tuning runs, evaluated on the development set, presented as empirical mean and standard deviation. All scores are percentage points. Highest score for each column is highlighted bold.

struggling to differentiate between *Medium*, *Topic*, and *Evidence*. In fact, the annotation guidelines intensively elaborate on a differentiation between these classes, which could hint at an inherent complexity of this task.

The second major source of errors seems to be that certain phrases are not recognized as roles at all by the system. In particular, there appears to be a disagreement between the system and the gold annotation as to which phrases belong to the *Message* and which do not. For instance, in the gold annotation "*Ich sage Ihnen eines, Herr Mützenich – das sage ich auch den Kollegen von Grünen und Linkspartei –: Wir diskutieren gerne über [Vermögenssteuern]. Jetzt müssen wir uns nur darum kümmern, dass es überhaupt noch eine wirtschaftliche Substanz gibt [...].*" the second and third sentence is part of the gold message span, but not in the prediction. Symmetric, in the prediction "*Fast alle mit Kindern unter drei Jahren arbeiten in Teilzeit, und – das sage ich ganz offen – es ist zu befürchten, dass sie aufgrund geringer Gehälter jetzt beruflich zurückstecken.*" the phrase after the parenthesis is not part of the gold role.

5.4 Testing Political Bias

As last part of my analysis, I want to provide some explorations on potential biases of my system along a political axis. The system might be used in more downstream tasks inferring information from German Bundestag debates, e.g., in a quantitative analysis comparing the speeches of the different parliamentary groups. Thus, to allow neutral inferences on such textual datasets, it is imperative to investigate potential imbalances in system performance, in particular between parliamentary groups, in order to avoid any unintended biases towards or against certain parliamentary groups.

For this, I am focusing on the system's accuracy, comparing the accuracies on the development set along the different parliamentary groups. In the following, I am referring with parliamentary groups to the groups (*Fraktionen*) that were represented in the 19th and 20th Bundestag. The development set speeches were pooled according to the parliamentary group the respective speaker is member of, as indicated by the GePaDe dataset.

To now infer differences in F1 score between the parliamentary groups, I performed parameter estimations through two separate regressions on the Match-Precision resp. Match-Recall on the development set. Here, the observations are the micro-averaged precisions resp. recalls on the speech events, that are used to compute Match-Precision resp. Match-recall. Since the distribution of the individual observations is highly bimodal (e.g., for each predicted speech event, micro-precision is either around 100 % or around 0 %) I chose to perform a Bayesian hierarchical beta-binomial regression. For instance, in the Match-Precision regression, for each observation the predicted-positive count is the number of trials, and the true-positive count is the number of successes. Now, instead of assuming a fixed success probability, the success probability is rather sampled, individually for each observation, from a high-level beta distribution corresponding to the respective parliamentary group. We are interested in inferring the shape parameters of these beta distributions. I particularly allowed in the prior for U-shaped beta distributions. Inference was conducted with PyMC[11] and the provided MCMC sampler.

The Bayesian models allow us to sample the mean parameter from the precision resp. recall beta distribution, and by taking the harmonic mean, we can visualize the posterior distributions of the Match-F1 score, for each parliamentary group respectively, as in Figure 3(a). Visually, we see how the estimates for F1 scores vary for each of the respective parliamentary group. The effect is most prominent between the SPD and LINKE group, where the model estimates the mean of F1 score for the particular group at 80.9 vs. 86.7 percent points. ($\Pr = 0.88$ for a difference of > 5 percent points be-

[11] https://www.pymc.io

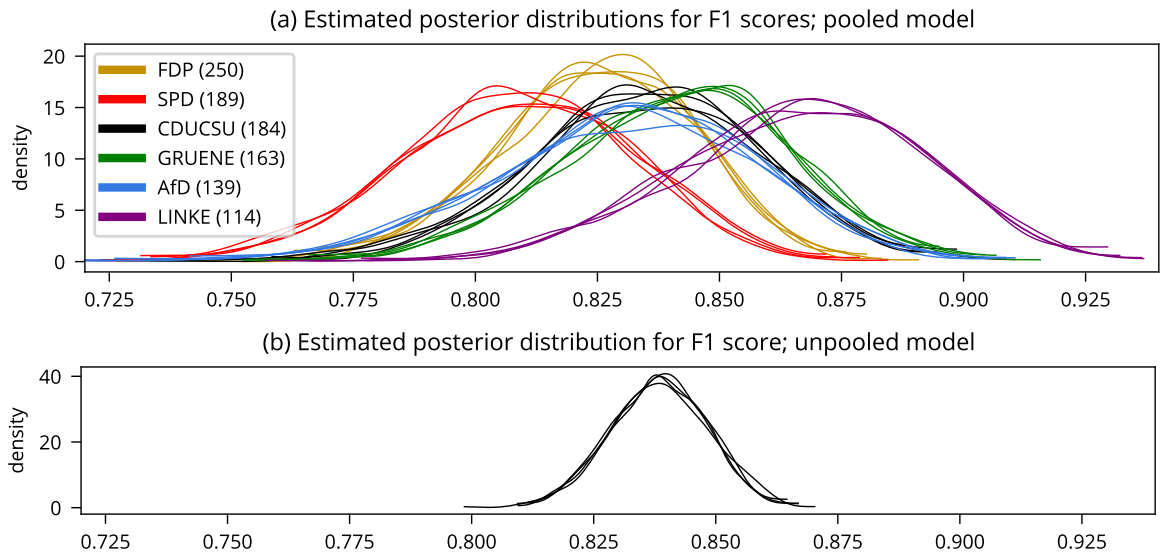

Figure 3: Estimated posterior distributions for the Match-F1 score on the development set. Each line corresponds to one of the four chains of the MCMC posterior draw. (a) Pooled model, where each parliamentary group has their own parameters and F1 posterior. The numbers in the legend indicate the number of observations per parliamentary group. (b) As a comparison, posterior predicted from the unpooled model, where every parliamentary group shares the same parameters, and thus F1 score is distributed identically over all groups.

tween the two groups.) However, it needs to be further examined whether this difference actually reflects a certain bias of the system towards certain textual phenomena or speech content, or whether this may be an effect of the random development split, where, e.g., the speeches of the LINKE group only randomly happen to be 'easy' ones.

Even from a statistical point of view, we should not overestimate this result. Especially in light of the low number of observations in the development split, we might possibly see the result of the model over-fitting the data, thus erroneously moving the F1 distributions apart. In fact, we can compare the previous pooled model with a unpooled model, where the distributions of the political groups are the same (Figure 3(b)). An estimation of their respective expected log pointwise predictive density shows that these are largely equal (-1097.0 ± 41.7 for the pooled model vs. -1084.0 ± 41.6 for the unpooled one, where higher is better), ranking no model clearly above the other (cf. Vehtari et al., 2017).

In total, we see some indication of a difference in system performance between the parliamentary groups, at least in the development dataset. Nonetheless, further investigations are required to verify if these imbalances remain stable even when moving to larger test sets. For this particular case at least, a model comparison indicates no significant statistical evidence of a performance imbalance along different parliamentary groups.

6 Conclusion

The present paper summarizes my submission for the Shared Task on Speaker Attribution SpkAtt-2023, specifically task 1 for attribution in parliamentary speeches. The system handles this task as a collection of token classification resp. sequence classification tasks, using BERT as base language model. Thus, the present system offers a simple BERT-based baseline model, which, despite its minimal architecture, provides a steady baseline. Even the variant based on the smaller $\text{GBERT}_{\text{Base}}$ model appears to have minimal performance losses, making it applicable to settings with less compute resources. In contrast, a domain adaptation through continued fine-tuning on a corpus of speeches from the German Bundestag led to no significant improvement. The error analysis indicates that the system is mostly struggling primarily with ambiguous 'edge cases,' where it appears to be not even entirely clear what the correct annotation would be. A quantitative comparison of the system's performance across the different parliamentary groups shows no strong evidence towards a potential imbalance. Overall, these results indicate the applicability of the system in further downstream analyses, e.g., in quantitative discourse studies of parliamentary debates.

References

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Simone Conia, Andrea Bacciu, and Roberto Navigli. 2021. Unifying cross-lingual semantic role labeling with heterogeneous linguistic resources. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–351, Online. Association for Computational Linguistics.

Simone Conia and Roberto Navigli. 2020. Bridging the gap in multilingual semantic role labeling: a language-agnostic approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1396–1410, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Simone Conia and Roberto Navigli. 2022. Probing for predicate argument structures in pretrained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4622–4632, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anton Ehrmanntraut, Leonard Konle, and Fotis Jannidis. 2023. LLpro: A literary language processing pipeline for German narrative texts. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, Ingolstadt, Germany. KONVENS 2023 Organizers. To be published.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. ArXiv: 2106.09685.

Leonard Konle and Fotis Jannidis. 2020. Domain and task adaptive pretraining for language models. In *Proceedings of the Workshop on Computational Humanities Research (CHR 2020)*, volume 2723 of *CEUR Workshop Proceedings*, pages 248–256, Amsterdam, the Netherlands.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. GitHub Repository.

Ines Rehbein, Fynn Petersen-Frey, Annelen Brunner, Josef Ruppenhofer, Chris Biemann, and Simone Paolo Ponzetto. 2023. Overview of the GermEval 2023 Shared Task on Speaker Attribution in Newswire and Parliamentary Debates. In *The GermEval 2023 Shared Task at KONVENS 2023*, Ingolstadt, Germany.

Florian Richter, Philipp Koch, Oliver Franke, Jakob Kraus, Lukas Warode, Fabrizio Kuruc, Stella Heine, and Konstantin Schöps. 2023. Open Discourse: Towards the first fully comprehensive and annotated corpus of the parliamentary protocols of the german bundestag. SocArXiv: dx87u.

Stefan Schweter and Alan Akbik. 2021. FLERT: Document-level features for named entity recognition. ArXiv: 2011.06993.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. ArXiv: 1904.05255.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. 2017. Practical bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.