Using Language Models for Classifying the Party Affiliation of Political Texts

Tu My Doan (✉)[0000−0002−9440−5847], Benjamin Kille[0000−0002−3206−5154], and Jon Atle Gulla[0000−0002−9806−7961]

Norwegian University of Science and Technology, Trondheim, Norway
{tu.m.doan, benjamin.u.kille, jon.atle.gulla}@ntnu.no

Abstract. We analyze the use of language models for political text classification. Political texts become increasingly available and language models have succeeded in various natural language processing tasks. We apply two baselines and different language models to data from the UK, Germany, and Norway. Observed accuracy shows language models improving on the performance of the baselines by up to 10.35 % (Norwegian), 12.95 % (German), and 6.39 % (English).

Keywords: Party Affiliation Classification · Political Text Representation · Language Models

1 Introduction

Neural Language Models (LMs)—large neural networks capturing patterns in extensive corpora of written language—have changed our abilities for automatically processing natural language. Abilities include text translation [8], code completion [15], and conversational agents [12]. There is, however, limited exploration of the use of LMs for political texts. These texts become increasingly available as organizations such as the European Union demand transparency.[1] Consequently, member states have adopted measures to facilitate access to political information. Many parliaments regularly publish their proceedings digitally. Citizens can read about the views and opinions of their representatives.

With the available data sources, we ask whether LMs can capture the inherent structure of political speech? We consider data from a set of nations and explore whether LMs can identify the speakers' party affiliation. More demanding use cases, such as identifying viewpoints, demand a large collection of annotated texts, which are lacking. Concretely, we formulate two research questions:

RQ_1 Do language models identify the party affiliation of political texts *more accurately* than a Naïve Bayes classifier?

RQ_2 Does language models' accuracy vary with different languages?

[1] The Treaty of the European Union states that "Every citizen shall have the right to participate in the democratic life of the Union. Decisions shall be taken as openly and as closely as possible to the citizen." (see http://data.europa.eu/eli/treaty/teu_2016/art_10/oj)

The remainder is structured as follow: Section 2 reviews related work. Section 3 outlines the data sets used for evaluation. Section 5 introduces the baselines and language models. Section 6 illustrates the results. Section 7 concludes.

2 Related Work

Classifying party affiliations takes a corpus of party-related texts and evaluates different predictors. Research on party classification has frequently used texts from the United States leading to a binary classification problem. For instance, Bei et al. [17] explore the use of Naïve Bayes (NB) and Support Vector Machines (SVM) on Congressional data. Dahllöf [4] uses SVMs to classify speeches of Swedish politicians. Wong et al. [16] focuses on identifying political leaning or voting preferences of Twitter users with data from the 2012 US. presidential election. The authors model the task as convex minimization. Rao and Spasojevic [10] use Recurrent Neural Networks (RNN) and word embeddings to detect political leaning of social media users. Again, the task was formed as binary classification (Democratic/Republican). Biessmann et al. [3] explore bag-of-word representation to predict whether texts come from government or opposition members. Høyland et al. [6] classify speeches in a multi-class setting with SVM. Baly et al. [2] use language models to classify news articles into left, center, and right. Kummervold et al. [7] fine-tuned a classifier on a balanced dataset of Norwegian Parliamentary speeches for party affiliation detection using Transformers and NB-BERT language model. Cases with a multi-party democracy represent a harder challenge than the binary classification into Democrat or Republican. We explore the use of language models for three such multi-class problems in the UK, Germany, and Norway.

3 Data

We consider three datasets of different languages (Norwegian, German, and English). First, we pre-process the data. Table 1 shows the datasets' composition. We split the data into training, validation, and test set (see Table 2).

3.1 Norwegian Parliamentary Speech Corpus (NPSC)

The Norwegian Language Bank at the National Library of Norway developed the NPSC [13] data set consisting of transcribed meeting recordings and speakers' meta data from 2017 and 2018. The recordings amount to 140 h of running speech, 65k sentences, and 1.2M words. We focus exclusively on the text data (speeches and metadata). As the average speech is 137 words long, we filtered out speeches with fewer than 150 words. To reduce the imbalance in this dataset, we decided to remove parties with less than 100 speeches. This resulted into a new dataset of total 3091 speeches of seven parties.

Table 1: Distribution of political speeches per party. We removed parties with fewer than 100 speeches of at least 150 words. ID refers to the party label. N refers to the initial number of speeches. M refers to the final number of speeches. Further, we show the proportion of speeches retained (% ret), and their distribution over all parties (% prop).

ID	Party	N	M	% ret	% prop
	Norwegian Parliamentary Speech Corpus (NPSC)				
–	Arbeidernes ungdomsfylking *(Workers' Youth)*	3	-	-	0.0
0	Arbeiderpartiet *(Norwegian Labour)*	2637	571	21.7	18.5
1	Fremskrittspartiet *(Progress Party)*	1444	632	43.8	20.4
2	Høyre *(Right Party)*	3216	977	30.4	31.6
3	Kristelig Folkeparti *(Christian Democrats)*	425	142	33.4	4.6
–	Miljøpartiet De Grønne *(Green Party)*	75	-	-	0.0
–	Rødt *(Red Party)*	30	-	-	0.0
4	SV – Sosialistisk Venstreparti *(Socialists)*	464	224	48.3	7.2
5	Senterpartiet *(Center Party)*	1090	351	32.2	11.4
6	Venstre *(Liberal Party)*	338	194	57.4	6.3
	Sum	9722	3091	31.8	100.0
	German Parliamentary Speech Corpus (GPSC)				
0	AFD (*Alternative for Germany*)	4437	2950	66.5	3.4
1	Bündnis 90/Die Grünen (*Green Party*)	23 975	13 789	57.5	15.9
2	CDU / CSU (*Christian Democrats*)	41 252	26 520	64.3	30.6
3	DIE LINKE (*Left Party*)	16 776	10 362	61.8	12.0
4	Fraktionslos (*without party affiliation*)	876	496	56.6	0.6
5	FDP (*Liberal Party*)	17 062	10 998	64.5	12.7
6	PDS (*Party of Democratic Socialism*)	1739	1066	61.3	1.2
7	SPD (*Social Democrates*)	29 497	20 396	69.1	23.6
–	not found	75	-	0.0	0.0
	Sum	135 689	86 577	63.8	100.0
	UK Parliamentary Debates Corpus (ParlVote)				
–	Alliance	13	-	-	0.0
0	Conservative	13 530	7915	58.5	41.4
1	Dup	578	269	46.5	1.4
–	Green	116	-	-	0.0
2	Independent	229	127	55.5	0.1
–	Independent-conservative	5	-	-	0.0
–	Independent-ulster-unionist	9	-	-	0.0
3	Labour	13 195	7557	57.3	39.5
4	Labourco-operative	784	426	54.3	2.2
5	Liberal-democrat	2864	1773	61.9	9.3
6	Plaid-cymru	338	167	49.4	0.9
–	Respect	6	-	-	0.0
7	Scottish-national-party	1436	756	52.6	4.0
8	Social-democratic-and-labour-party	189	128	67.7	0.7
–	Ukip	14	-	-	0.0
–	Uup	155	-	-	0.0
	Sum	33 461	19 118	57.1	99.5

3.2 German Parliamentary Speech Corpus (GPSC)

We use the data set created by Richter et al. [11] capturing the German parliament's speeches between 1949 and present. To establish a fair comparison, we extracted speeches from 2000 and later. The speeches contain some noise. First, the texts contained meeting minutes' page numbers. Second, the texts contains line breaks. We removed both obtain a better textual representation of the actual speech. We obtained a total of 135 689 speeches. We applied the pre-processing pipeline and retained speeches with at least 150 words of parties with at least 100 such speeches. The dataset has 86 577 speeches of eight parties.

3.3 UK Parliamentary Debates Corpus (ParlVote)

Abercrombie and Batista-Navarro [1] collected transcribed parliament records[2] between 7 May 1997 and 5 November 2019. The dataset[3] contains 34 010 speeches with information about debate ID, motion, title, and speakers' metadata (ID, name, political party, and votes). There are two versions: *ParlVote_full* (34 010 speeches) and *ParlVote_concat* (33 461 speeches of 1995 debates). We work with the latter—pre-processed subset of data used for down-streaming task (sentiment analysis), and consider only speeches, and party. Applying same strategy as with NPSC and GPSC, the final dataset has 19 118 speeches of nine parties.

Table 2: Summary of the split datasets for running experiments.

Dataset	Total items	Train	Validate	Test	# Parties
NPSC	3091	2318	193	580	7
GPSC	86 577	64 932	5411	16 234	8
ParlVote	19 118	14 338	1195	3585	9

4 Methods

We consider three types of classifiers. First, we discuss the baselines. Second, we introduce a selection of language models. Finally, we explore how these language models can be fine-tuned for the task at hand.

4.1 Baselines

We need baselines to assess the added value of LMs. We consider two baselines.

[2] https://www.theyworkforyou.com/
[3] https://data.mendeley.com/datasets/czjfwgs9tm/2

Majority Class represents a trivial choice. The baseline predicts the same label for all instances in the test set corresponding to the majority class in the training corpus. Consequently, the Majority Class baseline helps us to see whether the other approaches learn non-trivial pattern.

Naïve Bayes (NB) represents a more competitive baseline for comparison with the LMs. Naïve Bayes has been found to be a viable baseline for 'traditional' natural language processing tasks [17]. We use a TF-IDF representation and build a classifer with the auto generated vocabulary from `sklearn`[4].

4.2 Language Models

For Neural Nets (NN), we fine-tune classification models[5] for the task. We selected models that are either multi-lingual or based on texts of the needed language (English, German, Norwegian). We fine-tune the models for the classification task with the training data. Models are trained on NVIDIA A100 40GB and 80GB GPU. For finding hyperparameters for Transformer models, we explore with number of epoch max to 15, learning rates $\in$ {1e-5, 1e-4, 1e-3, 2e-5, 2e-4, 3e-5, 4e-5, 4e-4, 5e-5}, batch size $\in$ {32, 64} and max sequence length 512 for BERT and GPT-2 language model. Best hyperparameters are chosen based on the accuracy on validation set. Table 3 shows selected training hyperparameters for fine-tuning models.

BERT Introduced by Devlin et al. [5], BERT has been successfully achieving state of the art results for many NLP tasks such as question answering, text generation, and sentence classification. BERT is the contextual embeddings transformer-based model which is pre-trained on a huge corpus using two tasks: masked language model and next sentence prediction. The authors use WordPiece tokenization and a 30 000 token vocabulary. There are two standard configurations: $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$. In the scope of this work, we use variations of BERT for different languages.

- *bert-base-multilingual-cased* [5][6]—a multilingual Transformer model for 104 languages. We use this language model for all three languages in our experiments.
- *nb-bert-base* [7][7]—A Norwegian transformer language model owned by the National Library of Norway.

[4] We use the `MultinomialNB` classifier, remove stopwords (Norwegian/German/English), use n-grams from 1 to 4. We determine the best hyperparameter configuration with grid search over maximum number of features $\{30k, 50k, 100k\}$ and the learning rate $\alpha \in \{0.01, 0.1, 0.5, 1.0\}$. For the NPSC data, we use 30 000 features and $\alpha = 0.01$. For the GPSC data, we use 100 000 features and $\alpha = 0.1$. For the ParlVote data, we use 100 000 features and $\alpha = 0.01$.

[5] https://huggingface.co

[6] https://github.com/google-research/bert

[7] https://github.com/NBAiLab/notram

Table 3: Fine-tuning hyperparameters for transformer models using validation set. #EP refers to number of trained epochs. #BS refers to batch size and LR denotes learning rate.

Dataset	Model name	#EP	#BS	LR
NPSC	TM-mbert	11	64	5×10^{-5}
	TM-nb-bert-base	13	64	5×10^{-5}
	TM-norwai-gpt2	11	32	1×10^{-4}
	TM-nb-bert-base-weighted	15	32	4×10^{-5}
	TM-nb-bert-custom-lm	10	32	5×10^{-5}
	TM-nb-bert-weighted-custom-lm	8	32	4×10^{-5}
GPSC	TM-mbert	5	64	3×10^{-5}
	TM-bert-base-german-cased	5	64	3×10^{-5}
	TM-german-gpt2	1	32	2×10^{-4}
	TM-bert-base-german-cased-weighted	9	32	4×10^{-5}
	TM-bert-base-german-cased-custom-lm	4	32	4×10^{-5}
	TM-bert-base-german-cased-weighted-custom-lm	13	64	5×10^{-5}
ParlVote	TM-mbert	3	32	4×10^{-5}
	TM-bert-base-cased	4	64	3×10^{-5}
	TM-english-gpt2	8	32	2×10^{-4}
	TM-mbert-weighted	13	64	2×10^{-5}
	TM-mbert-custom-lm	3	32	3×10^{-5}
	TM-mbert-weighted-custom-lm	12	32	4×10^{-5}

– *bert-base-german-cased*[8]—a German BERT model developed by deepset.ai team in 2019.
– *bert-base-cased* [5][9]—A pretrained model on English language using a masked language modeling (MLM) objective.

GPT-2 is a large language model by Radford et al. [9] which is built on transformer decoder block. GPT-2 is trained on WebText dataset in the self-supervised way. The model has achieved state of the art results on many NLP task and is the key importance to the success of zero-shot task transfer. GPT-2 uses Byte-Level BPE tokenizer with extended vocabulary size to 50 257. There are various sizes for GPT-2 whereas the largest has 1542M parameters and 117M parameters for the smallest.

– *norwai-gpt2*[10] - A Norwegian pretrained transformer model which is in the process of training by NorwAI.

[8] https://huggingface.co/bert-base-german-cased
[9] https://huggingface.co/bert-base-cased
[10] https://www.ntnu.edu/norwai/new-language-models-in-norwai

- *german-gpt2*[11] – a language model for German owned by Bayerische Staatsbibliothek (Bavarian State Library).
- *english-gpt-2*[12] [9] – a transformer model pretrained on a very large corpus of English data in a self-supervised fashion.

4.3 Models Refinement

We apply various strategies to the original transformer fine-tuning models to improve the accuracy of the classifiers. We pick the model with the highest accuracy in each corpus for refining. First, to deal with the imbalanced data, we calculate class weight where classes with more data have less weights than their counterparts. Second, we continue training the LM on the within-task training data. Finally, we combine both methods to check the effect on the accuracy. Table 5 shows results for all refined models.

Balancing Training Data with Class Weights: All datasets that we consider are highly imbalanced thus providing a bigger challenge for us. We can expect that the models overfit for the majority classes while performing poorly for the minority classes. To tackle the issue, we estimate class weights[13] for unbalanced data and integrate that into *CrossEntropyLoss*. Similar grid search and fine-tuning strategy are done.

Training Language Model on Custom Dataset: To improve the transformer models, we follow the strategy from Sun et al. [14] by training LMs using within-task training data. We use all speech data in the training set, split them into proportion of 0.9 and 0.1 respectively for training and validating language model. To find the best training hypeparameters for language models, we do grid search for batch size ∈ {32, 64}, block size ∈ {128, 256}, learning rate ∈ {1e-5, 1e-4, 2e-5, 3e-5, 4e-5, 5e-5}, and maximum 10 000 training steps on small subset of data. Then, best parameters are used to train the language model with early stopping. Best checkpoint is selected based on evaluation loss. Later the transformer uses this language model for fine-tuning classifier.

5 Experiments

To answer our research questions, we define *accuracy* as our evaluation criterion. In other words, we measure the accuracy of both baselines, language models, and fine-tuned language models on all three datasets. Therein, we present the texts of the test set to all classifiers and check whether their predictions match the actual party-affiliations. We fine-tune the best-performing language model either

[11] https://huggingface.co/dbmdz/german-gpt2
[12] https://huggingface.co/gpt2
[13] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

Table 4: Parameters for training language models on within-task training data including max training steps (TS), batch size (BS), block size (BLS), and learning rate (LR)

Model name	Base model	Dataset	Parameters			
			TS	BS	BLS	LR
nb-bert-base-custom-ds	nb-bert-base	NLSP	$4k$	32	128	2×10^{-5}
bert-base-german-cased-custom-lm	bert-base-german-cased	GPSC	$100k$	64	256	1×10^{-4}
mbert-custom-ds	mbert	ParlVote	$13k$	64	256	1×10^{-4}

with weighting, customization, or both. We do not distinguish between members of the governing parties and opposition members. Subsequently, we can compare the accuracy for the models and languages. The data, methods, and evaluation protocols are publicly available.[14]

6 Results

Table 5 outlines the classifiers' overall performance on the three data sets. The trivial *Majority Class* baseline achieves the lowest accuracy. The Naïve Bayes classifier predicts the correct party for texts in about 12 to 13 in 20 cases. We observe that the language models outperform the Naïve Bayes classifier by up to 10.35 % (Norwegian), 12.95 % (German), and 6.39 % (UK). Thus, we can conclude that overall language models predict the party affiliation of political texts more accurately than the 'traditional' Naïve Bayes classifier. Still, class-specific performance varies among approaches. For all data sets we observe some classes that challenge all approaches. For instance, the class 6 in the German data set sees the lowest performance by all methods. Note, the *Majority Class* baseline performs perfectly for one class while failing all others.

Figure 1a shows the distribution of the difference in class-specific accuracy between the language models and the Naïve Bayes baseline. The horizontal line at 0 highlights the point where baseline and language model perform identically. Much of the distribution is to the right of the line indicating that the language models perform better than the baseline in most cases. In particular, the German data reveals a large proportion of cases beyond 50 %.

The difference in performance for *TM-mbert* shows the performance across language barriers. The accuracy varies marginally between 67.64 % (German) and 71.58 % (English). The superior performance in English could be the results of a majority of the training corpus being written in English.

Figure 1 shows the class-specific difference in performance of the best performing language model (*TM-nb-bert-weighted-custom-lm*) and the Naïve Bayes baseline for the Norwegian data set (other figures omitted due to space limitations). The cells show the difference in cases between the LM and the baseline.

[14] https://github.com/doantumy/LM_for_Party_Affiliation_Classification

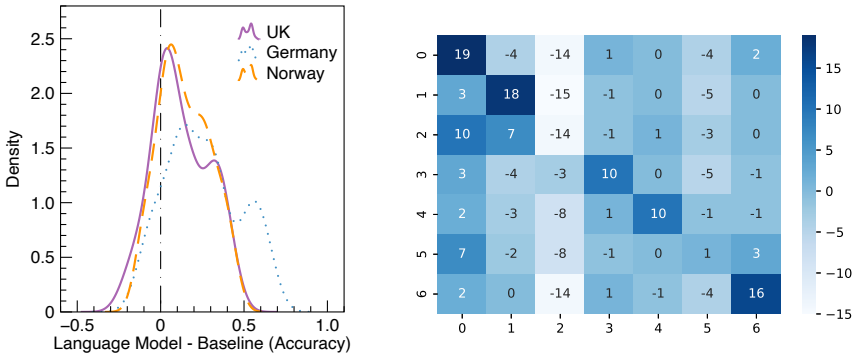
Table 5: Results overview. For each data set and method, we show the overall accuracy, the best class-specific performance, as well as the worst class-specific performance. There are three groups of models per data set. First group shows the baselines. Second group represents classifiers with different language models. Third group denotes the refinement classifiers.

Dataset	Method	Accuracy	Best Class Label	Worst Class Label
NPSC	Majority Class (baseline)	31.55	100.00 (2)	0.00 (not 2)
	NB (baseline)	62.93	86.89 (2)	13.51 (6)
	TM-mbert	68.79	84.70 (2)	21.43 (4)
	TM-nb-bert-base	68.97	78.99 (1)	37.84 (6)
	TM-norwai-gpt2	66.03	75.41 (2)	42.86 (4)
	TM-nb-bert-base-weighted	69.14	76.47 (1)	51.35 (6)
	TM-nb-bert-custom-lm	72.24	86.89 (2)	43.24 (6)
	TM-nb-bert-weighted-custom-lm	**73.28**	87.40 (1)	50.00 (4)
GPSC	Majority Class (baseline)	30.64	100.00 (2)	0.00 (not 2)
	NB (baseline)	61.70	88.10 (2)	1.00 (6)
	TM-mbert	67.64	81.98 (2)	51.26 (6)
	TM-bert-base-german-cased	72.26	85.08 (2)	49.75 (6)
	TM-german-gpt2	70.64	86.53 (2)	54.27 (6)
	TM-bert-base-german-cased-weighted	71.35	81.05 (0)	53.77 (6)
	TM-bert-based-german-cased-custom-lm	73.60	82.59 (2)	34.67 (6)
	TM-bert-based-german-cased-weighted-custom-lm	**74.65**	82.13 (0)	57.29 (6)
ParlVote	Majority Class (baseline)	41.39	100.00 (0)	0.00 (not 0)
	NB (baseline)	66.72	81.81 (0)	0.00 (4)
	TM-mbert	71.58	81.93 (3)	0.00 (2)
	TM-bert-base-cased	71.24	83.96 (0)	0.00 (4, 8)
	TM-english-gpt2	66.47	81.47 (0)	6.25 (4)
	TM-mbert-weighted	56.80	60.20 (3)	17.50 (4)
	TM-mbert-custom-lm	**73.11**	87.94 (0)	3.75 (4)
	TM-mbert-weighted-custom-lm	73.02	84.23 (0)	12.50 (4, 8)

The rows correspond to predicted classes, whereas the columns represent the actual values. The cells are color-coded for better visualization. The language model performs slightly worse on the majority class with label 2. Conversely, the language model assigns labels more accurately for all other classes.

The performances seem consistent for all three languages. In all three data sets, a language model achieves the best performance with accuracy between 73.11 to 74.65 percent. The Naïve Bayes baseline achieves less accurate score in the range 61.70 to 66.72 percent. The *mbert* model represents a special case due to its multi-lingual character. We applied it to all three scenarios. We observed the best performance for the English data (71.58 %) followed by the Norwegian (68.79 %) and the German (67.64 %) data. Language-specific models achieved higher accuracy for Norwegian (*TM-nb-bert-weighted-custom-lm* with 73.28 %) and German (*TM-bert-base-german-case-weighted-custom-lm* with 74.65 %).

Figure 2 shows the relation between the number of training instances and the model type with the classification accuracy. We computed the z-score of the number of training examples such that we can compare texts across lingual barriers. The plots show the data points, a linear regression, and the compatibility



(a) Histogram of Accuracy (b) Class-specific heatmap

Fig. 1: The left shows the performance differences between the language models and the Naïve Bayes baseline. The horizontal line at 0 highlights the point where baseline and language model perform identically. The right-hand side shows a heatmap of class-specific differences in accuracy between *TM-nb-bert-weighted-custom-lm* and the *Naïve Bayes* classifier.

region. The subplot on the bottom right compares the three types of models. We observe that all types of models perform better for classes with more training instances. This confirms findings for Swedish by Dahllöf [4]. The enhance language models, which were tuned with the training samples, perform best. The regular language models still perform better than the baseline. The classes with few training examples show a high level of variance independent of the model type. Consequently, we can deduce that having more training examples represents a valuable asset for political text classification.

7 Conclusions

In conclusion, we analyze the effectiveness of different language models for three languages (Norwegian, German, and English) in the problem of classifying political affiliation of authors. Research on the use of artificial intelligence and machine learning for political texts is still relatively fresh. This work encourages more efforts towards the use of language models and related resources for political texts. The results show us that language models give better accuracy in classifying all three languages (RQ_1). The difference in accuracy compared to the *majority class* baseline indicates that both the Naïve Bayes and the LMs have learned some meaningful patterns. Further, language models with refinement on the training data performed better than unrefined models. We have seen that language models benefit of large sets of training examples. Conversely, the performance of all classifiers for classes with few training instances remained poor. This suggests that having a domain-specific language model is going to

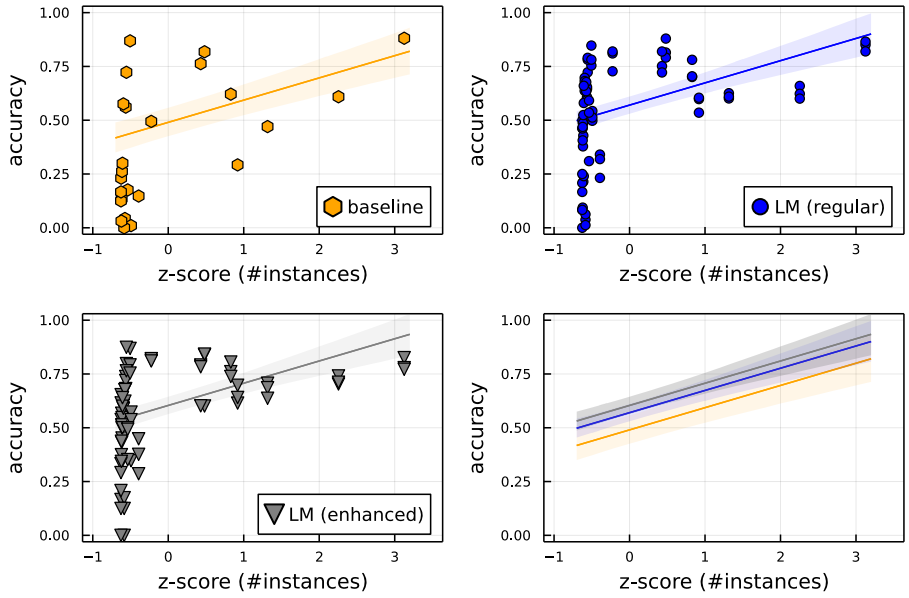


Fig. 2: Effect of the number of training instances and type of model on classification accuracy. For each of the model types (baseline, regular and enhanced language model), a figure shows the relation between the number of training instances (z-score) and the accuracy.

help improve the results of the task. The fact that all of the three data sets suffer from imbalance problem has raised the importance of building balanced and decent datasets for political research. The variance of *TM-mbert* across language barriers shows that the performance does not vary drastically (RQ_2).

As next steps, we will annotate a large corpus of political texts. Repeating the experiments with these additional resources ought to reveal whether more and better training data or more sophisticated, deep models promise better results. Besides, we plan to extend the experiment to further languages to verify that given a language model, the performance for party affiliation classification benefits. We will pay particular attention to languages which are under-resources such as Swedish, Danish, Finnish, Dutch, or Hungarian. Furthermore, we will carefully investigate errors made by the classifiers to better understand their deficiencies. With sufficient training data, we plan to create a LM specific to political speech. The data used for our experiments are publicly available. We hope that other researchers will join our efforts and replicate our experiment.

Acknowledgements This work is done as part of the Trondheim Analytica project and funded under Digital Transformation program at Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway.

References

1. Abercrombie, G., Batista-Navarro, R.: ParlVote: A Corpus for Sentiment Analysis of Political Debates. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 5073–5078. European Language Resources Association (2020), https://aclanthology.org/2020.lrec-1.624
2. Baly, R., Da San Martino, G., Glass, J., Nakov, P.: We Can Detect Your Bias: Predicting the Political Ideology of News Articles. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 4982–4991 (2020). https://doi.org/10.18653/v1/2020.emnlp-main.404
3. Biessmann, F., Lehmann, P., Kirsch, D., Schelter, S.: Predicting political party affiliation from text. PolText 2016 **14**, 14 (2016)
4. Dahllöf, M.: Automatic Prediction of Gender, Political Affiliation, and Age in Swedish Politicians from the Wording of their Speeches—A Comparative Study of Classifiability. Literary and Linguistic Computing **27**(2), 139–153 (2012). https://doi.org/10.1093/llc/fqs010
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805 (2018). https://doi.org/10.18653/v1/N19-1423
6. Høyland, B., Godbout, J.F., Lapponi, E., Velldal, E.: Predicting Party Affiliations from European Parliament Debates. In: Proceedings of the ACL 2014 workshop on language technologies and computational social science. pp. 56–60 (2014)
7. Kummervold, P.E., De la Rosa, J., Wetjen, F., Brygfjeld, S.A.: Operationalizing a national digital library: The case for a Norwegian transformer model. In: Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa). pp. 20–29 (2021), https://aclanthology.org/2021.nodalida-main.3
8. Luong, M.T., Kayser, M., Manning, C.D.: Deep Neural Language Models for Machine Translation. In: Proc. of the 19th Conf. Comp. Natural Lang. Learning. pp. 305–309 (2015). https://doi.org/10.18653/v1/K15-1031
9. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language Models are Unsupervised Multitask Learners. OpenAI blog **1**(8), 9 (2019)
10. Rao, A., Spasojevic, N.: Actionable and Political Text Classification using Word Embeddings and LSTM. arXiv preprint arXiv:1607.02501 (07 2016)
11. Richter, F., Koch, P., Franke, O., Kraus, J., Kuruc, F., Thiem, A., Högerl, J., Heine, S., Schöps, K.: Open Discourse (2020), https://doi.org/10.7910/DVN/FIKIBO
12. Serban, I., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building End-to-end Dialogue Systems using Generative Hierarchical Neural Network Models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 30 (2016)
13. Solberg, P.E., Ortiz, P.: The Norwegian Parliamentary Speech Corpus. arXiv preprint arXiv:2201.10881 (2022)
14. Sun, C., Qiu, X., Xu, Y., Huang, X.: How to Fine-tune BERT for Text Classification? In: China National Conf. on Chinese Comp. Ling. pp. 194–206. Springer (2019). https://doi.org/10.1007/978-3-030-32381-3_16
15. Wang, W., Shen, S., Li, G., Jin, Z.: Towards Full-line Code Completion with Neural Language Models. arXiv preprint arXiv:2009.08603 (2020)
16. Wong, F.M.F., Tan, C.W., Sen, S., Chiang, M.: Quantifying Political Leaning from Tweets, Retweets, and Retweeters. IEEE Transactions on Knowledge and Data Engineering **28**(8), 2158–2172 (2016)
17. Yu, B., Kaufmann, S., Diermeier, D.: Classifying Party Affiliation from Political Speech. Journal of Information Technology & Politics **5**, 33–48 (07 2008). https://doi.org/10.1080/19331680802149608