Predicting Policy Domains from Party Manifestos with BERT and Convolutional Neural Networks

Allison Koh
Centre for International Security
Hertie School
koh@hertie-school.org

Daniel Kai Sheng Boey
School of International and Public Affairs
Columbia University
daniel.boey@columbia.edu

Hannah Bechara
Data Science Lab
Hertie School
bechara@hertie-school.org

Abstract

Hand-labeled political texts are often required in empirical studies on party systems, coalition building, agenda setting, and many other areas of political science research. While hand-labeling remains the standard procedure for analyzing political texts, it can be slow, expensive, and subject to human error. Recent studies in the field have leveraged supervised machine learning techniques to automate the labeling process of political texts. We build on current approaches to label shorter texts and phrases in party manifestos using a pre-existing coding scheme developed by political scientists for classifying texts by policy domain and preference. Using labels and data compiled by the Manifesto Project, we make use of the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) with Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU) to seek the best model architecture to supplant manual coding of political texts. We find that our proposed BERT-CNN model outperforms other approaches for the task of classifying political texts by policy domain.

1 Introduction

During campaigns, political actors communicate their position on a range of key issues to signal campaign promises and gain favor with constituents. Identifying the political positions of political actors is essential to understanding their intended political actions. This is why policy preferences—or positions on specific policy issues expressed in speech or text—have been extensively analyzed within the relevant political science literature (Abercrombie et al., 2019; Budge et al., 2001; Lowe et al., 2011; Volkens et al., 2013). Methods employed to investigate the policy preferences of political actors include analysis of roll call voting, position extraction from elite studies or regular surveys, expert surveys, hand-coded analysis, and computerized text analysis (Debus, 2009). Studies that utilize political manifestos, electoral speeches, and debate motions often rely on the availability of machine-readable documents that are labeled by policy domain or policy preference.

Quantitative methods, especially in the field of natural language processing, have enabled the development of more scalable methods for predicting policy preferences. These advancements have enabled political scientists to analyze political texts and estimate their positions over time (Nanni et al., 2016; Zirn et al., 2016). To better understand the political positions of political actors, many social science researchers have turned to hand-labeling political documents, such as parliamentary debate motions and party manifestos. Much of the previous work on analyzing political texts relies on hand-labeling documents (Abercrombie and Batista-Navarro, 2018; Gilardi et al., 2009; Krause, 2011; Simmons and Elkins, 2004). Thus, the analysis of political documents in this field stands to benefit from automating the coding of texts using supervised machine learning. Most recently, neural networks and deep language representation models have been employed in state-of-the-art approaches to automatic labeling of political texts by policy preferences.

In this paper, we present a deep learning approach to classifying labeled texts and phrases in party manifestos, using the coding scheme and documents from the Manifesto Project (Volkens et al., 2019). We use English-language texts from the Manifesto Project Corpus, which divides party manifestos into statements—or *quasi-sentences*—that do not span more than one grammatical sentence. Based on the state-of-the-art deep learning methods for text classification, we propose using Bidirectional Encoder Representations from Transformers (BERT) combined with neural networks to

automate the task of labeling political texts. We compare our models that combine BERT and neural networks against previous experiments with similar architectures to establish that our proposed method outperforms other approaches commonly used in natural language processing research to predict policy domains and policy preferences. We identify differences in performance across policy domains, paving the way for future work on improving deep learning models for classifying political texts. To the best of our knowledge, we offer the most comprehensive application of deep language representation models incorporated with neural networks for document classification of political manifesto statements.

The rest of this paper is structured as follows. In Section 2, we provide a brief overview of the current state-of-the-art methods in the classification of political texts, focusing mainly on detecting policy domains and preferences. Section 3 goes into detail about the Manifesto Project Corpus. Section 4 then introduces our classification approach and provides important details of our models and evaluation approach. In Sections 5 and 6, we present our results and address some limitations of our system. Finally, Section 7 concludes our findings and presents a roadmap for future improvements.

2 Related Work

For the task of classifying political texts, many studies have concentrated on building scaling models for identifying the political positions of documents (Laver et al., 2003; Nanni et al., 2019; Proksch and Slapin, 2010). However, most of this seminal work in this area failed to consider the task of classifying texts by topic or policy area prior to detecting policy preferences associated with the topic. Over the past couple of years, several studies have addressed this gap in *opinion-topic identification* by classifying text data from political speeches, manifestos, and other documents by topic before predicting policy preferences (Glavaš et al., 2017; Zirn et al., 2016). With regards to party manifestos, the coding of policy preferences after dividing documents into topics could be expansive, pointing to the necessity of more complex models for text classification to take on this task. This is why recent studies have begun to utilize neural networks (Subramanian et al., 2018) and deep language representation models (Devlin et al., 2018) to address the computationally intensive task of classifying political texts into over thirty categories.

Against this background, this project closely follows the methods proposed by Abercrombie et al. (2019), who worked to detect the policy positions of UK Members of Parliament through natural language processing methods. Using motions and manifestos as data sources, the authors employed a variety of methods to predict the policy and domain labels of texts. Thereafter, they compared the predicted labels with the gold standard labels to produce F1 scores. For their proposed BERT model, Abercrombie et al. (2019) used a final softmax model and added CNN and max-pooling layers. Furthermore, they fine-tuned the results of the aforementioned BERT Model by training it first on the manifestos and then on the motions. The authors evaluated the predicted labels of each experimental model against the gold standard labels (i.e., when two annotators agree on the same labels) produced during the annotation process. Ultimately, they found that the use of BERT demonstrated 'state-of-the-art performance' on both manifestos and motions via supervised pipelines, with a Macro-F1 score of 0.69 for their best performing model, pointing to the effectiveness of this model in predicting policy preferences from political texts.

3 The Manifesto Project Corpus

The Manifesto Project Corpus[1] (Volkens et al., 2019) provides information on policy preferences of political parties from seven different countries based on a coding scheme of seven policy domains, under which 57 policy preference codes are manually coded. The Manifesto Project offers data that divides party manifestos into quasi-sentences, or individual statements which do not span more than one grammatical sentence. Quasi-sentences are then individually assigned to categories pertaining to policy domain and preference. The 57 policy preference codes refer to the position—positive or negative—of a party regarding a particular policy area. The 57 policy preference codes fall into a macro-level coding scheme comprising of eight policy domain categories[2].Hereafter, we refer to the policy preferences and policy domains as 'minor' and 'major' categories, respectively. In political science research, the Manifesto Project Corpus is particularly useful for studying party competition,

[1] manifesto-project.wzb.eu

[2] Each topic classification scheme includes a distinction for "non-categorized" texts

the responsiveness of political parties to constituent preferences, and estimating the ideological position of political elites. While the official classification of manifestos in this dataset has primarily relied on human coders, the investigation of automatically detecting policy positions of the text data is valuable for scaling up the classification of large volumes of political texts available for analysis.

Our final subset of all English-language manifestos comprises of 99,681 quasi-sentences. Tables 1 and 2 illustrate the distribution of English-language manifestos across countries and policy domains. To ensure that the ratio between policy domains remains consistent across policy domains in running our models, we applied a 70/15/15 split between training, validation, and test sets separately for the eight major categories and the 57 minor categories. Test and validation sets were sampled to have the identical class distribution of the training data.

Table 1: English language manifestos by policy domain

Topic	QSs	%
External Relations	6580	6.7
Freedom and Democracy	4700	4.8
Political System	10557	10.7
Economy	24757	25.2
Welfare and Quality of Life	30750	31.3
Fabric of Society	11099	11.3
Social Groups	9910	10.1

Note: Excludes "non-categorized" statements.

Table 2: English language manifestos by country

Country	QSs	%
United States	10819	10.9
South Africa	6423	6.5
New Zealand	28561	28.7
Ireland	25352	25.5
Great Britain	14839	14.9
Canada	3047	3.1
Australia	10370	10.4

4 Experimental Setup

BERT has proven successful in prior attempts to classify phrases and short texts (Devlin et al., 2018). We test two variants of BERT—one incorporating a bidirectional GRU model, and another incorporating CNNs. Between these two variants, we propose that BERT-CNNs are the state-of-the-art application of deep learning for classifying statements from political texts.

4.1 Bidirectional Encoder Representations from Transformers (BERT)

BERT's key innovation lies in its ability to apply bidirectional training of transformers to language modeling. This state-of-the-art deep language representation model uses a "masked language model", enabling it to overcome restrictions caused by the unidirectional constraint. Our experiments use the standard pre-trained BERT transformers as the embedding layer in our model. We make use of the BERT BASE uncased tokenizer, with the following parameters:

$$\text{BERT}_{\text{BASE}}\text{: (L=12, H=768, A=12, TotalParameters=110M)}$$

Since BERT is trained on sequences with a maximum length of 512 tokens, inclusive of start and end of sentence tokens, all quasi-sentences with more than 510 words were trimmed to fit this requirement. Pre-trained embeddings of the entire transformer body were frozen and not trained for the base models. We utilized the Hugging Face `transformers` library to run our BERT and other deep language representation models[3]. Model specifications and training times for our neural networks and deep language representation models are shown in Tables 3 and 4.

4.2 RoBERTa

The RoBERTa model was proposed by Liu et al. (2019) in a replication study that evaluates several approaches to augmenting the process of pre-training BERT models. The adjustments made to improve upon BERT include training the model longer, removing the model's objective of predicting the next sentence, training on longer sequences of text, and changing the pattern of masking texts applied in the This masked language model improves on the performance of BERT models in several downstream tasks. In this research, we fine-tune RoBERTa with a simple linear classifier on top, using the RoBERTa BASE tokenizer.

4.3 BERT with Gated Recurrent Units (GRU)

First proposed by Cho et al. (2014), Gated Recurrent Units use update gates and reset gates to solve

[3] https://huggingface.co/transformers/

Models	Text Representation	Layers	Epochs
CNN	GloVe Wikipedia w-emb	2 Convolutional Layers (1 per filter) 2 Max Pooling Layers 1 Dropout Layer 1 Linear Layer	100
BERT	Base BERT (uncased)	1 Linear Layer	10
RoBERTa	Base RoBERTa	1 Linear Layer	10
BERT-CNN	Base BERT (uncased)	2 Convolutional Layers (1 per filter) 2 Max Pooling Layers 1 Dropout Layer 1 Linear Layer	10
BERT-GRU	Base BERT (uncased)	1 Bidirectional GRU RNN Layer 1 Dropout Layer 1 Linear Layer	10

Table 3: Model specifications of neural networks and deep language representation models

Table 4: Training time (in seconds) for neural networks and deep language representation models for classifying political texts by *major* and *minor* policy domain

Model	8 topics	57 topics
CNN	559	672
BERT	4123	3883
RoBERTa	4120	4110
BERT-CNN	2177	2085
BERT-GRU	2564	4820

the vanishing gradient problems often encountered in applications of recurrent neural networks (Kanai et al., 2017). The update gate helps the model determine the extent to which past information is carried on in the model, whilst the reset gate determines the information to be removed from the model (Chung et al., 2014). It solves the aforementioned problem by not completely removing the new input, instead keeping relevant information to pass on to further subsequent computed states. In our analysis, we employ a multi-layer, bidirectional GRU model from PyTorch[4]. The results are subject to a dropout layer prior to classification via a linear layer.

4.4 BERT with Convolutional Neural Networks (CNN)

We incorporate CNNs with BERT using the same CNN architecture as our baselines (Table 3). The model utilizes the aforementioned BERT base, uncased tokenizer with convolutional filters of sizes 2 and 3 applied with a ReLu activation function. We use a 1D-max pooling layer, a dropout layer ($N = 0.5$) to prevent overfitting, and a Cross Entropy Loss function. We employ the model to classify policy domains ($N = 8$) and policy preferences ($N = 57$), each of which includes a category for quasi-sentences that do not fall into this classification scheme. A graphical representation of our model is shown in Figure 1.

4.5 Evaluation

We evaluate the performance of our proposed method against several baselines, which include:

- **Multinomial Naive Bayes** (Eyheramendy et al., 2003): This algorithm, commonly used in text classification, operates on the *Bag of Words assumption* and the assumption of *Conditional independence*.
- **Support Vector Machines (SVM)** (Tong and Koller, 2001): We used this traditional binary classifier to calculate baselines with the `SVC` package from `scikit-learn`[5], employing a "one-against-one" approach for multi-class classification.
- **Convolutional Neural Networks (CNN)** (Kim, 2014; LeCun et al., 1998): To run this deep learning model, originally designed for image classification, we first made use of pre-trained word vectors trained by GloVe, an unsupervised learning algorithm for obtaining vector representations for words (Pennington et al., 2014)[6].

[4] https://pytorch.org/

[5] https://scikit-learn.org/stable/

[6] See Table 8 in the appendix for detailed information on pre-trained word embeddings.

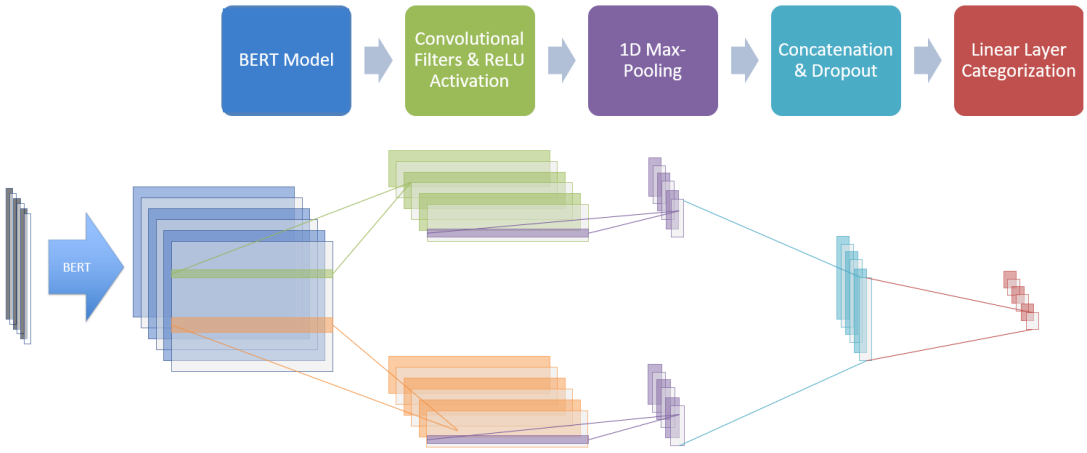

Figure 1: Graphical representation of the base BERT-CNN model to predict major policy domains.

To evaluate model fit, we utilized *accuracy* and *loss* as key metrics to compare performance of our CNN and BERT-GRU baseline against the BERT-CNN model. We calculated the *F1-score* for each model that we ran. In our results, we present both the Macro-F1 and Micro-F1 scores[7].

4.6 Architecture fine tuning

We tested different modifications of the CNN and BERT models as a robustness check on the performance of our base model for the task of political text classification. For the CNN models, we compared our base model to the following modifications:

- **Stemming and Lemmatization**: We test whether stemming or lemmatizing text in the pre-processing steps improves predictions using quasi-sentences from the Manifesto Project Corpus.
- **Dropout rates**: We decreased the dropout rate from 0.5 to 0.25 to determine whether fine-tuning dropout rates yield differences in performance. This is because we initially found that our models were overfitting.
- **Additional linear layer**: An additional linear layer was added prior to the final categorization linear layer to establish whether "deeper" neural networks generate improved predictions.
- **Removal of uncategorized quasi-sentences**: The results from our base models yield lower Macro-F1 scores due to the difficulty of correctly categorizing quasi-sentences that do not fall into any of the eight policy domains or 57 policy preference codes. We are thus interested in whether predictions improve if the uncategorized quasi-sentences are taken out of the data used for analysis.

For the BERT models, we compared our base model to the following modifications:

- **Training Embeddings**: For our base BERT models, all training of embeddings were frozen. In this modification, we enable the training of the embeddings to establish how training embeddings contributes to the performance of deep language representation models with this classification task.
- **Training models based on recurrent runs**: We trialed training the BERT models sequentially with different learning rates (LR = 0.001, 0.0005 and 0.0001) of 10 epochs each for a total of 30 epochs in aims to improve the performance of our neural networks and deep language representation models.
- **Large, cased tokenizer**: The BERT Large cased tokenizer was used instead of the BERT BASE uncased tokenizer employed in our base models.

5 Results

As shown in Table 5, the BERT-CNN model performed best for predicting both major and minor categories compared to the BERT-GRU model and

[7]The micro score calculates metrics globally, whilst the macro score calculates metrics for each label and reports the unweighted mean.

Category	Model	Test Loss	Test Acc.	Micro-F1	Macro-F1
Major	MNB	—	0.553	0.553	0.398
	SVM	—	0.578	0.578	0.460
	CNN	1.177	0.589	0.589	0.466
	BERT	1.379	0.502	0.502	0.363
	RoBERTa	1.350	0.514	0.515	0.360
	BERT-GRU	1.166	0.594	0.593	0.479
	BERT-CNN	**1.152**	**0.591**	**0.591**	**0.473**
Minor	MNB	—	0.385	0.385	0.154
	SVM	—	**0.463**	**0.463**	**0.299**
	CNN	2.136	0.454	0.454	0.273
	BERT	2.457	0.376	0.376	0.177
	RoBERTa	2.621	0.354	0.354	0.136
	BERT-GRU	2.216	0.432	0.432	0.239
	BERT-CNN	**2.098**	0.448	0.448	0.260

Table 5: Baseline, CNN and masked language models run with base model specifications as detailed in Table 3

CNN baseline. However, our SVM baseline outperformed the neural network models for predicting minor categories. We believe that the shortcomings of our neural networks and deep language representation models for this text classification task are due to computational limitations in specifying the number of epochs in training. We also observed overfitting in our models. For instance, Figure 3 illustrates that training accuracy of our CNN model increased at the cost of validation accuracy. However, this was not the case for deep language representation models classifying texts by minor categories. Overall, our results demonstrate that, between the two BERT models, the BERT-CNN model demonstrates superior performance against bag-of-words approaches and other models that utilize neural networks.

CNN and BERT Modifications

Comparing modifications to our CNN models, our results suggest that the base model outperforms most alternative model specifications. As outlined in Table 6, reducing the dropout rate to 0.25 improved the model on some indicators marginally. As expected, the removal of uncategorized quasi-sentences yielded improvements in predictions, with a significantly higher Macro-F1 score compared to other model specifications. Based on these results, future work should focus on how model predictions of uncategorized quasi-sentences can be improved, given their random nature.

While we observed some improvements with modifications to the CNN model, we find that our base BERT models performed best compared to other fine-tuned modifications to model architecture. The results of our base BERT model and alternative model specifications are shown in Table 7. Even though it is possible that our base BERT model is best for this classification model, our results could also indicate the presence of over-fitting or the lack of sufficient training available given the low number of epochs.

6 Limitations and Analysis

As shown in Figure 2, we observed overfitting with our major policy domain classification models. Despite employing changes and modifications to our models, including varied dropout rates, architecture fine-tuning and different learning rates, we did not find any variants of the models employed in analysis that would yield significant improvements in performance. We posit that potential improvements on these issues could be resolved by employing transfer learning and appending our sample of English-language manifestos with other political documents, such as debate transcripts.

In contrast, as shown in Figure 3, we observed underfitting in some of our minor policy domain classification models. Our classifier could benefit from employing transfer learning and appending our sample of manifesto quasi-sentences with other political texts, especially for policy domains with relatively fewer quasi-sentences to train on. It is also important to note that, compared to the more computationally intensive neural networks and deep language representation models, our Multi-

Model	Change	Test Loss	Test Acc.	Micro-F1	Macro-F1	Epochs
CNN	Base model	1.177	**0.589**	**0.589**	0.466	100
	Lemmatized text	**1.174**	0.585	0.585	0.460	100
	Stemmed text	1.213	0.577	0.576	0.448	100
	Dropout = 0.25	1.177	**0.589**	0.588	**0.467**	100
	Additional layer	1.180	0.586	0.586	0.462	100
	Removing uncategorized QSs	**1.136**	**0.596**	**0.595**	**0.535**	100

Table 6: Comparing results of modifications to CNN base models for predicting major policy domains

Model	Change	Test Loss	Test Acc.	Micro-F1	Macro-F1	Epochs
BERT-GRU	Base model	**1.152**	**0.594**	**0.593**	**0.479**	10
	Training emb	1.163	0.592	0.592	**0.479**	10
	Recurrent runs, training	1.234	0.582	0.581	0.459	30
	Large, uncased	1.172	0.592	0.591	0.469	10
BERT-CNN	Base model	1.166	**0.591**	**0.591**	**0.473**	10
	Training emb	1.167	0.587	0.587	0.458	10
	Recurrent runs, training	**1.157**	0.589	0.589	0.468	30
	Large, uncased	1.192	0.580	0.580	0.450	10

Table 7: Comparing results of modifications to BERT base models for predicting major policy domains

nomial Bayes and SVM baselines did not perform significantly worse. In fact, for the minor categories, the SVM yielded superior performance in some metrics compared to that of the neural network models. Notwithstanding the lack of training of certain models, this may suggest that increasing the model complexity and consequently the computational power required may not necessarily lead to increased model performance.

Substantially lower Macro-F1 scores across all models point to mixed performance in classification by category. As shown in Figure 4, we observe high variation in the performance of our classifiers between categories. However, we observe poor performance in classifying quasi-sentences that do not belong to one of the seven policy domains. For our BERT-CNN model, the easiest categories to predict were "welfare and quality of life", "economy", and "external relations". The superior performance of predicting the first two categories is not particularly surprising, as a substantial number of quasi-sentences in our sample of English-language party manifestos are attributed to these topics. As shown in Table 1, 30,750 quasi-sentences are attributed to the "welfare and quality of life" category and 24,757 quasi-sentences are attributed to the "economy" domain.

In contrast, the relatively superior performance of predicting the "external relations" category is surprising. Out of our total sample of $n_{\text{sentences}} = 99,681$, only $6,580$ documents are attributed to this category[8]. The performance of our classifier with this underrepresented policy domain could be attributed to a variety of possible explanations. One possible explanation is the presence of distinct features, such as topic-unique terms, that do not exist in other categories. Future work on classification of political documents that fall under this category would benefit from looking into features that might establish which policy domains perform better than others with the BERT-CNN classifier.

7 Conclusion

In this paper, we trained two variants of BERT—one incorporating a bidirectional GRU model, and another incorporating CNNs. We demonstrate the superior performance of deep language representation models combined with neural networks to classify political domains and preferences in the Manifesto Project. Our proposed method of incorporating BERT with neural networks for classifying English language manifestos addresses issues of reproducibility and scalability in labeling large volumes of political texts. As far as we know, this is the most comprehensive application of deep language

[8]Some of the policy preferences coded under "External Relations" include foreign special relationships, anti-imperialism, peace, military, internationalism, and European community/union.

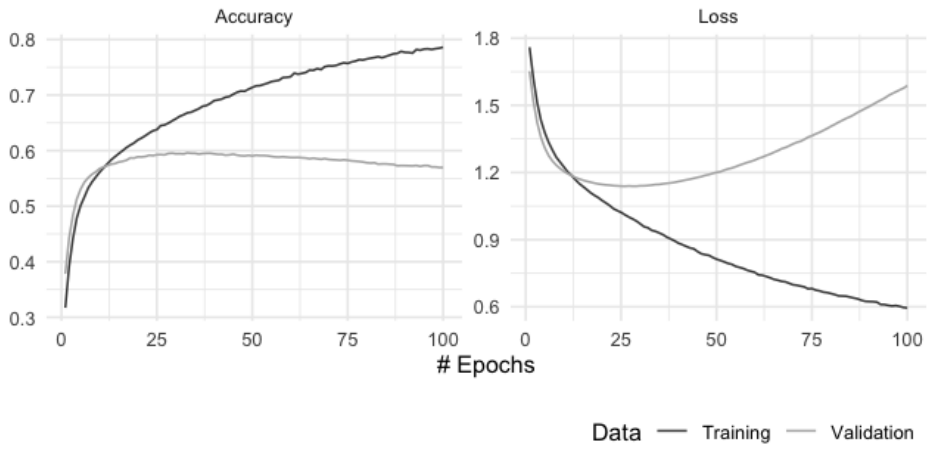

Figure 2: An illustration of overfitting in our CNN model for classifying manifesto quasi-sentences by major policy domain

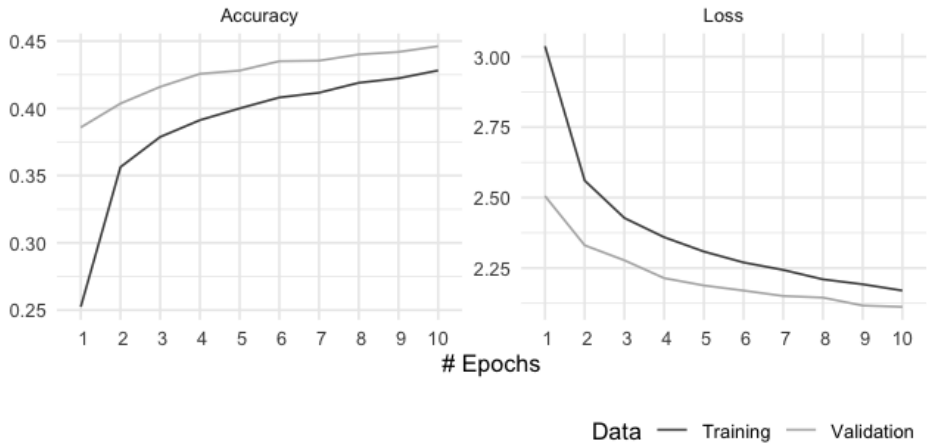

Figure 3: Training and validation metrics for the BERT-CNN model on English language manifestos on minor policy domains

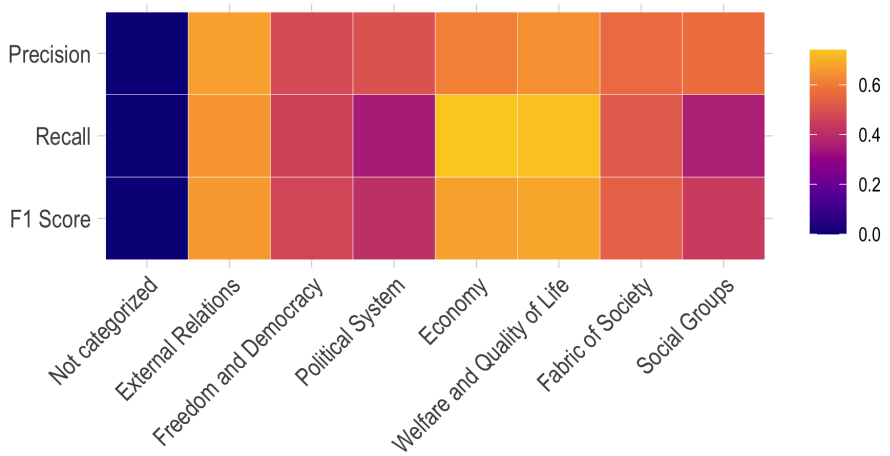

Figure 4: Average precision, recall, and Macro-F1 scores by major category across all models

representation models and neural networks for classifying statements from political manifestos.

We find that using BERT in conjunction with Convolutional Neural Networks yields the best predictions for classifying English language statements parsed from party manifestos. However, our proposed BERT-CNN model requires further fine-tuning to be effective in providing acceptable predictions to improve on less computationally intensive classifiers of fine-grained policy positions. As expected, our proposed approach and baselines perform better for classifying major policy domains over minor categories. We also observe differences in performance between categories. Among the major policy domains, the categories that performed best were "welfare and quality of life", "economy", and "external relations". The superior performance of the latter category is surprising because it makes up a relatively small proportion of quasi-sentences in the Manifesto Project Corpus.

There are several avenues for future work on neural networks and deep language representation models for the automatic labeling of political texts. For instance, investigating the features of individual categories that demonstrate superior performance could shed light on how we could incorporate additional features of texts to improve model performance. This area of research would also benefit from better understanding how we can filter out texts that do not fall into a particular classification scheme. Knowledge on how these issues could be resolved to improve model performance would allow for extensions in the application of deep learning models to the classification of political texts.

References

Gavin Abercrombie and Riza Batista-Navarro. 2018. A sentiment-labelled corpus of hansard parliamentary debate speeches. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Ed. by Darja Fišer, Maria Eskevich, and Franciska de Jong. Miyazaki, Japan: European Language Resources Association (ELRA)*.

Gavin Abercrombie, Federico Nanni, Riza Theresa Batista-Navarro, and Simone Paolo Ponzetto. 2019. Policy preference detection in parliamentary debate motions. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 249–259.

Ian Budge, Hans-Dieter Klingemann, Andrea Volkens, Judith Bara, Eric Tanenbaum, et al. 2001. *Mapping policy preferences: estimates for parties, electors, and governments, 1945-1998*, volume 1. Oxford University Press on Demand.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Marc Debus. 2009. *Estimating the Policy Preferences of Political Actors in Germany and Europe: Methodological Advances and Empirical Applications*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Susana Eyheramendy, David D Lewis, and David Madigan. 2003. On the naive bayes model for text categorization. In *International workshop on artificial intelligence and statistics*, pages 93–100. PMLR.

Fabrizio Gilardi, Katharina Füglister, and Stéphane Luyet. 2009. Learning from others: The diffusion of hospital financing reforms in oecd countries. *Comparative Political Studies*, 42(4):549–573.

Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. 2017. Cross-lingual classification of topics in political texts. Association for Computational Linguistics (ACL).

Sekitoshi Kanai, Yasuhiro Fujiwara, and Sotetsu Iwamura. 2017. Preventing gradient explosions in gated recurrent units. In *Advances in neural information processing systems*, pages 435–444.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Rachel M Krause. 2011. Policy innovation, intergovernmental relations, and the adoption of climate protection initiatives by us cities. *Journal of urban affairs*, 33(1):45–60.

Michael Laver, Kenneth Benoit, and John Garry. 2003. Extracting policy positions from political texts using words as data. *American political science review*, 97(2):311–331.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Will Lowe, Kenneth Benoit, Slava Mikhaylov, and Michael Laver. 2011. Scaling policy preferences from coded political texts. *Legislative studies quarterly*, 36(1):123–155.

Federico Nanni, Goran Glavas, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2019. Political text scaling meets computational semantics. *arXiv preprint arXiv:1904.06217*.

Federico Nanni, Cäcilia Zirn, Goran Glavaš, Jason Eichorst, and Simone Paolo Ponzetto. 2016. Topfish: topic-based analysis of political position in us electoral campaigns.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, page 1532–1543.

Sven-Oliver Proksch and Jonathan B Slapin. 2010. Position taking in european parliament speeches. *British Journal of Political Science*, 40(3):587–611.

Beth A Simmons and Zachary Elkins. 2004. The globalization of liberalization: Policy diffusion in the international political economy. *American Political Science Review*, pages 171–189.

Shivashankar Subramanian, Trevor Cohn, and Timothy Baldwin. 2018. Hierarchical structured model for fine-to-coarse manifesto text analysis. *arXiv preprint arXiv:1805.02823*.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Andrea Volkens, Judith Bara, Ian Budge, Michael D McDonald, and Hans-Dieter Klingemann. 2013. *Mapping policy preferences from texts: statistical solutions for manifesto analysts*, volume 3. OUP Oxford.

Andrea Volkens, Onawa Lacewell, Pola Lehmann, Sven Regel, Henrike Schultze, and Annika Werner. 2019. The manifesto data collection. manifesto project (mrg/cmp/marpor). *Berlin: Wissenschaftszentrum Berlin für Sozialforschung (WZB)*.

Cäcilia Zirn, Goran Glavaš, Federico Nanni, Jason Eichorts, and Heiner Stuckenschmidt. 2016. Classifying topics and detecting topic shifts in political manifestos.

A Additional Information on Baselines

For the first two methods, the Multinomial Naive Bayes Model and the Support Vector Machines, the `TfidfVectorizer` from `sklearn` was employed. This method makes use of term frequency - inverse document frequency weighting to remove terms that are present commonly but carry very little information (e.g. stopwords).

A.1 Multinomial Naive Bayes Model

As a baseline, we used a multinomial naive Bayes algorithm, commonly used in text classification. The assumptions of this model includes:

- Bag of Words: Position does not matter
- Conditional Independence: Feature probabilities are independent given the class.

The Naive Bayes Model is quick and provides a baseline for the other classification techniques.

A.2 Support Vector Machines

Support Vector Machines (SVM) seek the most optimal decision boundaries by creating hyperplanes that separate the training data (Tong and Koller, 2001). The aim of the separating hyperplane or the set of hyperplanes is to maximise the distance between the nearest training data points of any class (i.e. functional margin). Whilst SVMs are traditionally binary classifiers, `scikit-learn`'s package `SVC` employs a "one-against-one" approach for multi-class classification. Where a SVM is trained based on data from two classes and repeated for each relationship with each other class present. Since there are eight policy domains (including unclassified), there will be 28 distinct SVMs created.

We trained SVMs on both datasets with four different kernels:

- Linear kernel: $< x, x' >$
- Polynomial kernel: $(\gamma < x, x' > +r))^d$
- Radial basis function kernel: $(\gamma||x - x'||^2)$
- Sigmoid kernel: $(\tanh(\gamma < x, x' > +r))$

	GloVe 6B
Tokens	6 billion
Dimension	300
Vocabulary size	400 thousand
Cased?	No

Table 8: Details of GloVe pre-trained vectors utilized

A.3 Convolutional Neural Networks

Convolutional Neural Networks are neural networks that utilize layers that contain convolving filters that help to aggregate data into multiple layers (LeCun et al., 1998). Whilst it was originally designed for image classification, it has also been utilized for Natural Language Processing purposes - semantic parsing, sentence modeling, sentence classification, etc (Kim, 2014).

In our model, we first made use of pre-trained word vectors trained by GloVe, an unsupervised learning algorithm for obtaining vector representations of words(Pennington et al., 2014). Specifically, we chose pre-trained vectors trained on a corpus of 1.6 billion tokens from a 2014 Wikipedia dump.

Filter-sizes of 2 and 3 were used with 100 2D convolutional filters each. After a single convolutional layer per filter size, each of the layers are fed into the soft-max activation functions. Thereafter, a single max-pooling layer was utilized per filter. The outputs from the corresponding max-pooling layers were then concatenated and passed through a dropout layer. Lastly, the results were passe d through a linear layer to predict the different classifications.

In all models (the current and the following models), the Adam Optimizer was utilized with a Cross Entropy Loss function. The latter is a combination of a logistic softmax and a negative log likelihood loss functions, useful for classification problems with multiple classes.