

gerber at Touché: Ideology and Power Identification in Parliamentary Debates 2024

Notebook for the Touché Lab at CLEF 2024

Christian Gerber¹

¹University of Tübingen, 72070 Tübingen, Germany

Abstract

In democratic countries, national parliaments shape laws and policies through deliberative processes that reflect underlying political ideologies and power structures. This paper presents a system developed for the Ideology and Power Identification Shared Task to classify parliamentary debates into categories indicative of ideology and power dynamics. Using a Convolutional Neural Network (CNN) architecture enhanced with hyper-parameter optimisation, this system processes multilingual data from the ParlaMint corpus. Key preprocessing steps include cleaning, tokenisation and conversion of text into integer sequences. The CNN model consists of embedding, convolutional, max-pooling and dense layers with a sigmoid activation function for binary classification. Our evaluation, based on precision, recall and F1 score, shows that the model successfully classifies ideology and power dynamics in parliamentary debates, achieving an average F1 score of 0.676 for power identification and 0.632 for political orientation. These results demonstrate the potential of the model for analysing complex parliamentary discourse.

Keywords

CNN, Ideology and Power Identification, NLP, Touché

1. Introduction

Since the dawn of civilisation, politics has been a fundamental part of human society. From early tribal councils to the complex governmental structure of modern society, politics has shaped the way societies are organised, governed and led. Throughout history, political systems have evolved to meet the changing needs of society and to adapt to new challenges and opportunities. Without politics it will be difficult to maintain order in a large society, therefore politics is essential for the stability, security and development of a country, providing a structure through which decisions are made and power is distributed [1]. It influences every aspect of our lives, from the laws that are made, the resources that are distributed, the education, welfare and security of its citizens. Understanding this political communication and the presentation of political speakers is vital for a functioning society. These speeches usually consist of indirect speech and are quite complex. Nevertheless, it is important to analyse parliamentary debates in order to gain critical insight into how political ideologies and power dynamics influence legislative outcomes and, more importantly, the lives of citizens. Parliamentary debates are an essential part of democratic processes. Politicians and their associated parties serve as the voice of their constituents by expressing political ideologies, negotiating and making decisions [2]. This leads to the characterisation of a nation's political landscape. These debates provide a rich corpus for analysis, reflecting the political climate and the ideologies of each individual speakers. However, the inherent complexity and subtlety of political language, combined with the volume of textual data generated during parliamentary proceedings, presents significant challenges for computational analysis. Traditional approaches or simple approaches like analysing short and direct tweets f.e. via "X", often do not capture the nuanced expressions of political ideology and party affiliation in parliamentary discourse. The shared task [3] focuses on identifying two variables associated with speakers in a parliamentary debate: their political ideology and whether they belong to a governing party or a party

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

✉ christian.gerber@student.uni-tuebingen.de (C. Gerber)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

in opposition by using machine learning and natural language process. Offering a large corpus to train on. These two tasks were achieved by using a CNN model described in the third chapter.

2. Background

The analysis of political discourse has a rich history, with early studies focusing on the rhetorical strategies used by political representatives. While early work [1][2] emphasised the importance of persuasive elements used by politicians to shape public opinion and policy, recent advances include the integration of machine learning techniques into political discourse analysis. A study by Abercrombie and Batista-Navarro [4] provides a large and systematic review of sentiment and ideology detecting analysis in parliamentary debates. They discuss different approaches, ranging from sentiment analysis, classification and position scaling to the analysis of political speeches, in order to highlight the strengths and limitations of these methods. They found out, that within the overall area of sentiment analysis in political detection, there are eight tasks. For example, emotion analysis, agreement and alignment detection and most interesting for this paper: ideology and party affiliation detection. These tasks have been tackled using a wide range of approaches, from supervised to unsupervised machine learning methods, including neural networks. The use of convolutional neural networks (CNNs) has become increasingly popular due to their ability to capture complex patterns within textual data. Several comprehensive reviews provide insights into how deep learning models, such as CNNs, are applied to various text classification tasks, including sentiment analysis and political text analysis [5][6]. In addition, the analysis of X's tweets has become increasingly popular in recent years. COVID-19 tweets were analysed by Aslan et al. [7]. They used FastText Skipgram for information extraction, a convolutional neural network (CNN) model for feature extraction, and an adaptive optimisation algorithm (AOA) for feature selection. In Dehghani and Yazdanparast's paper [8], they present several machine learning and deep learning models to analyse the sentiment of Persian political tweets. They applied Gaussian Naive Bayes, Gradient Boosting, Logistic Regression, Decision Trees, Random Forests, as well as a combination of CNN and LSTM to classify the polarities of tweets. The results showed, that the CNN-LSTM model had the highest classification accuracy, showcasing the effect of CNN models in political text analysis.

3. System Overview

The following section provides a detailed description of the system developed to identify ideology and power in a political speech. The aim of these two tasks is to classify parliamentary debates into categories that reflect ideology or power dynamics. First, the input data is pre-processed and then CNN's local feature extraction is used to convert textual information into numerical vectors. This description outlines the components, resources and methods used to build and fine-tune the model. The software used for this implementation is Python with libraries including TensorFlow, Keras, Scikit-learn, Keras-tuner and others used for machine learning and data processing. The system uses a Convolutional Neural Network (CNN) architecture and employs various hyperparameter optimisation techniques to improve performance. Finally, the model is evaluated based on precision, recall and F1 score.

3.1. Dataset

The data for this task comes from ParlaMint [9], a multilingual comparable corpora of parliamentary debates. A selection of speeches was collected and made available as training set [3]. The data in the training set was sampled to reduce potential confounding variables (e.g. speaker identity) and provided in tab-separated text files. The fields in the data include:

- id: Unique ID for each text.
- speaker: Unique ID for each speaker, allowing multiple speeches from the same speaker.

- `sex`: Binary/biological sex of the speaker (Female, Male, Unspecified/Unknown).
- `text`: Transcribed text of the parliamentary speech, potentially including line breaks and other special sequences.
- `text_en`: Automatic translation of the text to English, which may be empty for speeches originally in English or missing for some non-English speeches.
- `label`: Binary/numeric label indicating political orientation (0 for left, 1 for right) or power identification (1 for opposition, 0 for coalition/governing party).

For the system described in this paper, only the fields 'id', 'speaker', 'text' and 'label' were used. Furthermore, the training data provided 29 different parliaments, including Austria, Belgium, Denmark and many more.

3.2. Data Preprocessing

Data preprocessing is the process of preparing raw data before it is used to build machine learning models, and involves several steps. As parliamentary speeches are sometimes published in different ways, there is a lot of inconsistency and redundancy, which makes data cleaning necessary. The following preprocessing steps were applied:

- Remove line breaks and non-alphabetic characters
- Convert text to lower case

At its core, CNN uses a "tokeniser" from Tensorflow [10], which is used to convert sequences of integers from the input data, with a vocabulary size limited to 10,000 words.

3.3. Model Architecture

Convolutional Neural Networks (CNNs) are a type of artificial neural network that learns directly from data. A CNN is a feed-forward network that can extract features from data with convolutional structures [5]. As a result of the convolutional layer, the input data is filtered and a feature map is created that illustrates the particular attributes associated with the data points [6]. CNN is able to detect local and deep features from text by using layers to automatically learn their hierarchies. The CNN model in this paper classifies the input text data and classifies it into two categories: for political orientation, 0 is left and 1 is right, and for power identification, 0 indicates coalition (or ruling party) and 1 indicates opposition. Figure 1 shows the proposed deep learning architecture using CNN. First, the embedding layer converts the pre-tokenised integer sequences into dense vectors of fixed size in an embedding matrix. One-dimensional convolution (Conv1D) is used for feature extraction. The next layer is max-pooling, which reduces the network parameters, resulting in a faster training process and easier handling of overfitting problems [11]. As a final layer, the dense layer with a sigmoid activation function is used to generate predictions. The model is built using the Adam optimiser and the binary cross-entropy loss function. Precision, recall and F1-score are used as evaluation metrics.

3.4. Training Process

The model is trained on the processed text data, split into training and validation sets to evaluate performance during training. The ratio of the splits were 80-20. During the process, the training data is carefully prepared to ensure no overlaps are made between speakers in the training and validation sets. The training was done on a GPU-enabled environment.

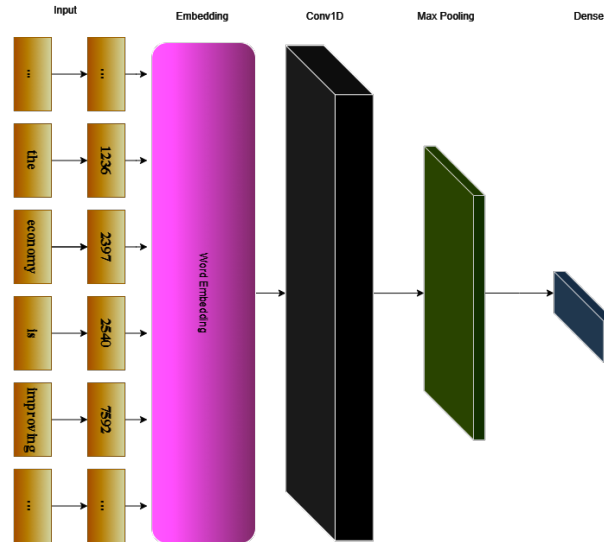


Figure 1: Architecture of the deep learning model

3.5. Hyperparameter Tuning

There are eight different parameters used in the neural network, each of which can have a different value. For the epoch 4, 5, 6, 7, 8, 9, 10, 11, 12 were considered and for the batch size 2, 4, 8, 16, 32 and 64. To find the best parameters and optimise the performance of the CNN, the Bayesian optimisation tuner from KerasTuner [12] was used. Bayesian optimisation is a sequential design strategy for optimising complex models where the decision-making process is not easily interpretable [13]. Basically, the goal was to optimise for validation accuracy. This was achieved by trying different combinations of words in the vocabulary (`max_nb_words`), embedding dimensions (`embedding_dim`), sequence length (`max_sequence_length`), number of convolutions (`num_conv_layers`), number of filters (`num_filters`) and kernel size (3, 5, 7). The Bayesian optimisation would be run twice per combination with a maximum of 10 trials. The best parameters for the number of epochs and the batch size were tested manually. These values are summarised in Table 1.

Table 1: Hyperparameters used in Bayesian Optimization

Parameter	Values
epochs	4, 5, 6, 7, 8, 9, 10, 11, 12
batch size	2, 4, 8, 16, 32, 64
max_nb_words	5000, 10000, 15000, 20000
embedding_dim	50, 100, 150, 200, 250, 300
max_sequence_length	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000
num_conv_layers	1, 2, 3
num_filters	32, 64, 96, 128, 160, 192, 224, 256
kernel_size	3, 5, 7

In the end the model worked best under these condition:

- Epochs: 8
- Batch size: 32
- max_nb_words: 10,000
- embedding_dim: 250
- max_sequence_length: 600
- num_conv_layers: 1

- num_filters: 192
- kernel_size: 3

4. Results

4.1. Evaluation Metrics

To evaluate the performance of the model, measures of precision, recall and F1 scores were used for the two different tasks. The first task was to identify the ideology of the speaker's party and the other task was to identify the power. Depending on the identification task, they are calculated based on the confusion matrix, which has four values:

- True Positives (TP): The number of correctly identified speeches from "right" ideology/"governing" party (0)
- True Negative (TN): The number of correctly identified speeches from "left" ideology/"opposition" party (1)
- False Positive (FP): The number of incorrectly identified speeches from the "right" ideology/"governing" party (0) when it is actually from the "left" ideology/"opposition" party (1)
- False Negative (FN): The number of incorrectly identified speeches from the "left" ideology/"opposition" party (1) when it is actually from the "right" ideology/"governing" party

Precision indicates what proportion of predicted positives are actually Positive.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

Recall measures the proportion of Positives that are correctly classified

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

F1-score is a number between zero and one that represents the harmonic mean of precision and recall

$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

4.2. Evaluation

This section presents the results of each dataset in each task. The parameters used to train the deep learning model, as discussed in section 2.5 on hyperparameter tuning, were determined by Bayesian optimisation. The results were evaluated using the submission system provided by TIRA [14] and were published by Touché [3]. Table 2 shows the results of the Power Identification task and Table 3 shows the results of the Political Orientation task. The highest scores are highlighted in green and the lowest in red. The overall F1 scores are 0.6758 for political power and 0.6322 for political orientation. These scores indicate a moderate level of accuracy in the models, reflecting their ability to classify political ideology and power. For the identification of speaker power, the model had the highest precision for Hungary (0.8776), meaning that it had the highest proportion of true positives out of all positive predictions. Conversely, Ukraine had the lowest precision (0.5135). Recall was also highest for Hungary (0.8694), meaning that the model was able to successfully identify most of the true instances of political power in speeches. Again, Ukraine had the lowest score (0.5138). The highest F1 score is also found in Hungary (0.8727). This indicates a good balance between precision and recall, while the lowest F1 score in Ukraine (0.4756) indicates significant room for improvement in both precision and recall. This shows that the model was able to identify the political power of a speaker quite well in countries like Hungary, Turkey and Galicia, while it had difficulties with speakers in Ukraine, Italy and Bosnia and Herzegovina.

Table 2: Political Power scores

Overall F1_Orientation	0.6322		
Parliament	Precision	Recall	F1_score
Austria (at)	0.6859	0.6842	0.6827
Bosnia and Herzegovina (ba)	0.8150	0.5267	0.5081
Belgium (be)	0.6001	0.5989	0.5951
Bulgaria (bg)	0.6615	0.6607	0.6557
Czechia (cz)	0.6356	0.6368	0.6360
Denmark (dk)	0.6523	0.6389	0.6350
Spain (es)	0.7329	0.7117	0.7164
Catalonia (es-ct)	0.7994	0.8112	0.8042
Galicia (es-ga)	0.8696	0.8593	0.8630
Basque Country (es-pv)	0.7446	0.7413	0.7422
Finland (fi)	0.6161	0.6065	0.5975
France (fr)	0.7149	0.7056	0.7095
Great Britain (gb)	0.7268	0.7211	0.7237
Greece (gr)	0.7783	0.6872	0.6836
Croatia (hr)	0.6760	0.6334	0.6305
Hungary (hu)	0.8776	0.8694	0.8727
Italy (it)	0.6103	0.5556	0.5161
Latvia (lv)	0.6750	0.6198	0.6286
The Netherlands (nl)	0.6525	0.6391	0.6411
Poland (pl)	0.7699	0.7690	0.7694
Portugal (pt)	0.6676	0.6688	0.6582
Serbia (rs)	0.7982	0.7137	0.7303
Slovenia (si)	0.6168	0.5985	0.5814
Turkey (tr)	0.8372	0.8411	0.8375
Ukraine (ua)	0.5135	0.5138	0.4756

The results for the orientation tasks, as shown in Table 3, show that the model achieved the highest prediction for the Turkish dataset in terms of precision (0.8404), recall (0.8423) and F1 score (0.8410). Conversely, the model gave the lowest results for Latvia in terms of precision (0.5301) and recall (0.5083). The lowest F1 score (0.4456) is for the recognition of orientation for speakers from Bosnia and Herzegovina. According to the results, the CNN model performed better for countries such as Turkey, Poland and Spain and less accurately for countries such as Bosnia and Herzegovina, Latvia and Croatia. This suggests that there is a need to improve the model in these countries.

Table 3: Political Orientation scores

Overall F1_Orientation	0.6322		
Parliament	Precision	Recall	F1_score
Austria (at)	0.6785	0.6204	0.6034
Bosnia and Herzegovina (ba)	0.5795	0.5084	0.4456
Belgium (be)	0.6163	0.5642	0.5442
Bulgaria (bg)	0.6227	0.6168	0.6188
Czechia (cz)	0.5333	0.5496	0.5223
Denmark (dk)	0.5653	0.5632	0.5630
Spain (es)	0.7811	0.7795	0.7738
Catalonia (es-ct)	0.6619	0.6594	0.6551
Galicia (es-ga)	0.7726	0.7605	0.7643

Parliament	Precision	Recall	F1_score
Finland (fi)	0.5587	0.5496	0.5392
France (fr)	0.6499	0.5799	0.5806
Great Britain (gb)	0.7591	0.7620	0.7591
Greece (gr)	0.7228	0.7267	0.7196
Croatia (hr)	0.5749	0.5302	0.5123
Hungary (hu)	0.7286	0.6717	0.6898
Italy (it)	0.6447	0.6136	0.6020
Latvia (lv)	0.5301	0.5083	0.4858
The Netherlands (nl)	0.5940	0.5920	0.5928
Poland (pl)	0.8376	0.6740	0.7159
Portugal (pt)	0.6909	0.6873	0.6885
Serbia (rs)	0.7470	0.6113	0.6408
Slovenia (si)	0.7280	0.5997	0.5839
Turkey (tr)	0.8404	0.8423	0.8410
Ukraine (ua)	0.7994	0.6964	0.7315

5. Conclusion

This paper presents a CNN model, developed for the Touché Lab at CLEF 2024, that can be used to identify the ideology and power of a speaker in parliamentary debates. Through careful data pre-processing and the application of hyperparameter optimisation techniques, the model achieves a satisfactory level of accuracy. The model was evaluated on both tasks by TIRA, using datasets from different countries. The results indicate that the model performs moderately well, with an overall F1 score of 0.6758 for power identification and 0.6322 for political orientation. For power identification, the CNN model performed best for the Hungarian dataset with an F1 score of 0.8727 and for ideology identification it was the Turkish dataset with an F1 score of 0.8410. However, the performance of the model in regions such as Ukraine and Bosnia and Herzegovina shows that there is room for improvement. This suggests a need for further refinement, possibly through more specific data pre-processing or the integration of additional contextual data. The use of monolingual or multilingual pre-trained models could help to achieve these improvements. This could potentially be achieved by using monolingual or multilingual pre-trained language models. Google's BERT or mBERT architecture has typically been trained on a large corpus, and there are a variety of writing styles in the corpus, as well as many topics (e.g. science, novels, news). Multilingual or monolingual language models, such as BERT or mBERT, can capture the semantics and meaning of sentences in a language. Therefore, these models are used in data pre-processing for word embedding, resulting in text vectorisation that can be used as input to a neural network. In addition, other layers can be added to the CNN model, such as LSTM layers. [8]. Future work could also focus on expanding the dataset, improving the model architecture and exploring additional features to further improve classification performance.

In conclusion, this research represents a fundamental step towards more automated analysis of parliamentary debates, paving the way for deeper insights into the ideological and power dynamics that shape legislative outcomes. The methods and results presented here contribute to the broader discourse on political communication and its computational analysis, and highlight the potential for further innovation in this important area.

Acknowledgments

References

- [1] P. Chilton, *Analysing Political Discourse: Theory and Practice*, Analysing Political Discourse: Theory and Practice, Routledge, 2004. URL: <https://books.google.de/books?id=un1buuNipQIC>.
- [2] J. Charteris-Black, *Analysing Political Speeches: Rhetoric, Discourse and Metaphor*, Bloomsbury Publishing, 2018. URL: <https://books.google.de/books?id=1fhGEAAQBAJ>.
- [3] J. Kiesel, Ç. Çöltekin, M. Heinrich, M. Fröbe, M. Alshomary, B. D. Longueville, T. Erjavec, N. Handke, M. Kopp, N. Ljubešić, K. Meden, N. Mirzakhmedova, V. Morkevičius, T. Reitis-Munstermann, M. Scharfbillig, N. Stefanovitch, H. Wachsmuth, M. Potthast, B. Stein, Overview of Touché 2024: Argumentation Systems, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024)*, Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2024.
- [4] G. Abercrombie, R. Batista-Navarro, Sentiment and position-taking analysis of parliamentary debates: a systematic literature review, *Journal of Computational Social Science* 3 (2020) 245–270. URL: <https://doi.org/10.1007/s42001-019-00060-w>. doi:10.1007/s42001-019-00060-w.
- [5] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: Analysis, applications, and prospects, *IEEE Transactions on Neural Networks and Learning Systems* PP (2021) 1–21. doi:10.1109/TNNLS.2021.3084827.
- [6] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, S. S. Iyengar, A survey on deep learning: Algorithms, techniques, and applications, *ACM Comput. Surv.* 51 (2018). URL: <https://doi.org/10.1145/3234150>. doi:10.1145/3234150.
- [7] S. Aslan, S. Kızıloluk, E. Sert, Tsa-cnn-aoa: Twitter sentiment analysis using cnn optimized via arithmetic optimization algorithm, *Neural Computing and Applications* 35 (2023) 10311–10328. URL: <https://doi.org/10.1007/s00521-023-08236-2>. doi:10.1007/s00521-023-08236-2.
- [8] M. Dehghani, Z. Yazdanparast, Political sentiment analysis of persian tweets using cnn-lstm model, 2023. URL: <https://arxiv.org/abs/2307.07740>. arXiv:2307.07740.
- [9] T. Erjavec, M. Ogrodniczuk, P. Osenova, N. Ljubešić, K. Simov, A. Pančur, M. Rudolf, M. Kopp, S. Barkarson, S. Steingrímsson, Çöltekin, J. de Does, K. Depuydt, T. Agnoloni, G. Venturi, M. Calzada Pérez, L. D. de Macedo, C. Navarretta, G. Luxardo, M. Coole, P. Rayson, V. Morkevičius, T. Krilavičius, R. Dargís, O. Ring, R. van Heusden, M. Marx, D. Fišer, The parlamint corpora of parliamentary proceedings, *Language resources and evaluation* 57 (2022) 415–448. doi:10.1007/s10579-021-09574-0.
- [10] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, Tensorflow: A system for large-scale machine learning, 2016. arXiv:1605.08695.
- [11] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, *J Big Data* 8 (2021) 53.
- [12] D. Yogatama, N. A. Smith, Bayesian optimization of text representations, 2015. arXiv:1503.00693.
- [13] J. Snoek, H. Larochelle, R. P. Adams, Practical bayesian optimization of machine learning algorithms, 2012. arXiv:1206.2944.
- [14] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), *Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023)*, Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/

978-3-031-28241-6_20.