Ideology and Power Identification in Parliamentary Debates

Medved Fran, Šarić Dominik, Topić Karla

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
fran.medved@fer.hr, dominik.saric@fer.hr, karla.topic@fer.hr

Abstract

This research focuses on political ideology detection in parliamentary speeches using advanced natural language processing techniques. The dataset, derived from the ParlaMint project, includes speeches from 5 european countries. We propose the Recursive Transformer (RTransformer) and Recursive Linear Model (RLinearModel) for this task. By creating two versions of data, not processed and processed, we also analyze the impact of preprocessing. Our findings show that RTransformer outperform baseline models (SGD, RNN, and Transformer) in predicting political orientation, especially with preprocessing. The RTransformer achieves best accuracy and F1 scores on the processed data. This study shows how effective recursive models are at capturing the semantic nuances and key phrases within text. The models provide great tools for political analysts and researchers to efficiently process and interpret parliamentary speech data.

1. Introduction

In modern democratic societies, debates within national parliaments play a crucial role, not only shaping the lives of citizens within a country but often exerting influence on a global scale. These debates serve as a fundamental platform for political discourse, shaping public policy and reflecting the ideological and power dynamics within the political landscape.

However, the language used in parliamentary speeches is often complex and indirect, posing significant challenges for computational analysis. Traditionally, political scientists and analysts have manually analyzed and classified parliamentary speeches. They carefully review and interpret each speech to determine the speaker's political orientation and affiliations. Although this method is thorough, it is very time-consuming and cannot keep up with the large amount of data produced in modern parliamentary sessions. However, recent advancements in natural language processing (NLP) provide promising tools to address this challenge, particularly through Recurrent Natural Networks (RNNs) and Transformer models.

This research leverages the strengths of recursive Transformer and recursive Linear models to predict ideological orientation in parliamentary debates. The baseline models include Linear Classifier with Gradient Descent (SGD), RNN, and Transformer models, which will be compared with a newly proposed Recursive Transformer (RTransformer) model and Linear (RLinearModel) model. These models process text in sequences and convert them into embeddings. The RTransformer model uses an encoder to process the sequences, while the RLinearModel uses a linear function.

By automating the identification of political ideology, we can provide political analysts and researchers with powerful tools to process and interpret large volumes of speech data efficiently.

2. Related Work

Iyyer et al. (2014) applied recursive neural networks (RNNs) to detect ideological bias in text by using the hierarchical nature of language. Their model constructs a parse tree where each word in a sentence is represented by a vector, and these vectors are recursively combined to form phrase and sentence-level representations. This approach allows the model to capture ideological bias from the composition of words and phrases. Building on the same foundational idea of using parse trees to capture hierarchical structures, our research utilizes the recursive approach with a Transformer model rather than the RNN. This allows us to harness the power of self-attention mechanisms in Transformers, which can weigh the importance of each word in the context of the entire sequence (Iyyer et al., 2014).

Baly et al. (2020) investigated how to predict political ideology in news articles by using both Long Short-Term Memory Networks (LSTMs) and Bidirectional Encoder Representations from Transformers (BERT). They faced a significant challenge: models often picked up on the writing styles of specific news outlets instead of the political biases within the articles themselves. To overcome this, they introduced adversarial media adaptation and triplet loss pre-training. These techniques helped the models focus more on the ideological content rather than just stylistic elements (Baly et al., 2020).

Kulkarni et al. (2018) proposed a multi-view model for detecting political ideology in news articles by incorporating cues from the title, link structures, and content. This attention-based model significantly outperformed the state-of-the-art models by incorporating network structures and content cues. Their approach underlines the importance of using diverse features beyond textual content alone to enhance the detection of ideological bias (Kulkarni et al., 2018).

These studies collectively demonstrate significant advancements in the field of political ideology detection, each contributing unique methodologies and datasets that enhance our understanding of ideological bias in text. Building on these foundations, our research aims to leverage the strengths of recursive transformers to classify ideology in parliamentary debates.

3. Methodology

3.1. Dataset Description

The dataset for this study is derived from the ParlaMint project, a multilingual and comparable corpus of parliamentary debates (ParlaMint, 2021). Each entry in the dataset includes the transcribed speech text in English and a binary label indicating political orientation (0 for left-wing, 1 for right-wing). The model is trained on five separate documents, each containing speeches in a single language from Croatia, Great Britain, the Czech Republic, Denmark, and Estonia. By training each model on data from one specific folder, the performance metrics are calculated and then averaged to obtain the final results. Additionally, the dataset is divided with 60% used for training, 20% for validation to optimize hyperparameters, and the remaining 20% for testing.

3.2. Preprocessing

Preprocessing steps were applied to the dataset to enhance model performance. Two versions of the dataset were created: one with stopwords removed and another with stopwords retained. This dual approach enables a comparison of the models to assess the impact of preprocessing. Text normalization was performed by converting all text to lowercase to ensure consistency. For stopword removal, the spaCy library was utilized, which efficiently identifies and eliminates stopwords from the text (Honnibal and Montani, 2017).

3.3. Linear Classifier with Gradient Descent

The Linear Classifier with Gradient Descent is a supervised learning model used for classification tasks. It employs the Stochastic Gradient Descent (SGD) algorithm to optimize the log loss (logistic loss) function, finding the best decision boundary that separates data points of different classes. For the baseline in this study, the Linear Classifier with Gradient Descent is employed to classify the political orientation of parliamentary speeches.

3.4. Recurrent Neural Network

The Recurrent Neural Network (RNN) used in this study is designed to classify the political orientation of parliamentary speeches. This bidirectional RNN processes input sequences in both forward and backward directions, allowing it to capture the full context of each word. The model consists of four RNN layers with a hidden dimension of 600, followed by a ReLU activation and two fully connected layers. The final output is transformed into a probability using the sigmoid function.

RNN provides a robust baseline for comparing the performance of more advanced models such as Transformers and the proposed Recursive Transformer model.

3.5. Transformers

The Transformer model is designed to handle sequential data, making it highly effective for natural language processing tasks. Unlike traditional RNNs, Transformers use self-attention to process all tokens in a sequence simultaneously, capturing long-range dependencies more efficiently.

The model includes an embedding layer to convert input sequences into vectors and positional encodings to maintain word order. It consists of multiple Transformer encoder layers with multi-head self-attention and feed-forward networks. The output is pooled and passed through fully connected layers to produce a final probability using the sigmoid function. This Transformer model provides a strong baseline for classifying the political orientation of parliamentary speeches since our own proposed model will be based on the transformers architecture.

3.6. Proposed Model: Recursive Transformer

The Recursive Transformer (RTransformer) is the proposed model for classifying the political orientation of parliamentary speeches. This model builds on the standard Transformer architecture, incorporating recursion to handle hierarchical structures within the data more effectively.

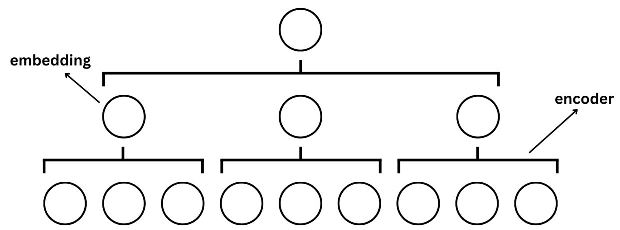

Figure 1: Architecture of the Recursive Transformer. This is an example for a row (text) with 9 words and kernel = 3. Circles represent embeddings of words.

This architecure is shown in Figure 1. The entire text from a single row in the dataset is treated as one input, with each word in the text being converted into an embedding (first level of circles in Figure 1), providing a dense representation of the word's meaning. Data is processed in batches with padding to ensure uniform input lengths. For each position t, the kernel captures a phrase of k word embedding. Word embeddings are based on a subset of pre-trained GloVe word representations from the Podium library. (Podium, 2020) The Transformer encoder then processes this phrase to produce a phrase-level representation:

$$h_t = \text{TransformerEncoder}(x_{t:t+k})$$

where $x_{t:t+k}$ denotes the sequence of k word embeddings starting at position t. The stride s determines the next starting position for the kernel, summarizing information in overlapping windows as it moves through the text. For subsequent layers, these phrase-level representations are recursively encoded:

$$h_{t+1} = \text{TransformerEncoder}(h_t)$$

Each recursive step reduces the sequence length by summarizing information. Finally, once the sequence is reduced to a single representation, a linear transformation followed by a sigmoid function is applied to generate the final classification output:

$$\hat{y} = \sigma(W_{\text{class}} \cdot h_{\text{final}} + b)$$

where W_{class} is the weight matrix for the classification layer, h_{final} is the final hidden state of the Transformer encoder, and σ is the sigmoid function. The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

The model is trained using the binary cross-entropy with logits loss function (`BCEWithLogitsLoss`), which combines a sigmoid layer and the binary cross-entropy loss in one single class. The loss function is defined as:

$$L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))]$$

where y_i is the true label, z_i is the raw output (logit) of the model, and N is the number of samples.

By processing the entire text and reducing the sequence length through recursive steps, the RTransformer can capture long-range dependencies and hierarchical structures within the data, providing a robust solution for political orientation classification.

3.7. Proposed Model: Recursive Linear

To compare the effectiveness of the RTransformer, a Recursive Linear Model (RLinearModel) was also implemented. The methodology for the RLinearModel follows the same process as the RTransformer, with the key difference being the use of a linear model instead of a Transformer encoder at each step. This substitution allows direct comparison between two models, highlighting the impact of using a Transformer encoder versus a linear model in capturing hierarchical structures and long-range dependencies in the text.

4. Results

The results in Table 1 show the performance for baseline models (SGD, RNN and Transformer) and proposed models (RTransformer and RLinearModel) at predicting the political ideology of speeches for not processed. Similarly, Table 2 presents the performance of the same models on the processed data.

Table 1: Results for test data which is not processed

Model	Acc	Prec	Rec	F1
SGD	69.1	71.2	72.1	60.4
RNN	65.6	83.8	31.6	41.3
Transformer	71.2	76.8	69.4	60.6
RTransformer	77.0	81.8	64.9	62.4
RLinearModel	64.2	62.2	54.6	58.2

Table 2: Results for test data which is processed

Model	Acc	Prec	Rec	F1
SGD	70.1	73.4	76.4	63.8
RNN	71.6	88.8	38.2	46.2
Transformer	78.2	82.8	72.8	65.8
RTransformer	81.8	87.2	71.4	68.3
RLinearModel	71.4	76.8	61.3	68.2

In Table 1, the RTransformer achieves the highest accuracy (77.0%) and F1 score (62.4%), indicating its robustness even without preprocessing. Among the baseline models, the Transformer shows a balanced performance with an accuracy of 71.2% and an F1 score of 60.6%, while the RNN, despite having a high precision, struggles with a low recall (31.6%), resulting in a lower F1 score.

Table 2 reveals a significant improvement in model performance after preprocessing the data. The RTransformer stands out again, with an accuracy of 81.8% and an F1 score of 68.3%, demonstrating the effectiveness of preprocessing for this model. The RLinearModel also shows improved performance, achieving an accuracy of 71.4% and an F1 score of 68.2%. The baseline Transformer and RNN model also benefits from preprocessing, with an improved accuracy and higher precision. However, RNN's recall remains low (38.2%), resulting in a correspondingly low F1 score.

Also, our approach of separating the dataset into folders by country proved to be effective. Initially, when the data was combined, the models exhibited lower scores. By training each model on data from specific countries, we ensured that the models could learn country-specific political orientations more effectively.

The comparative analysis between the non-processed and processed data results clearly indicates the positive impact of preprocessing. All models achieve substantial improvements in all metrics after preprocessing, with the RTransformer showing the best overall performance. The RLinearModel doesn't surpass the RTransformer, but still shows notable improvements and performs better than the RNN while being just below the Transformer model. This demonstrates the RTransformer's effectiveness in capturing hierarchical structures and long-range dependencies within the data, validating its suitability for the task.

5. Conclusion

In this research, we explored the classification of political ideology in parliamentary speeches using advanced natural language processing techniques. By leveraging the strengths of baseline models and developing a Recursive Transformer (RTransformer) and Recursive Linear Model (RLinearModel), we created a model that is capable of accurately identifying the political leanings of speakers based on their speech content.

The results from our experiments show that our proposed models are effective. The RTransformer performed well in terms of accuracy and F1 scores on the processed data, outperforming all other models. This highlights the significant impact of data preprocessing and demonstrates the benefits of using hierarchical processing to manage the complex structures in parliamentary speeches. The RLinearModel, while not surpassing the RTransformer, still showed notable improvements and positioned its performance between the RNN and Transformer models.

In conclusion, this study highlights the potential of Recursive Transformer in the domain of political ideology classification. By automating the identification of political orientation in parliamentary debates, our models give political analysts and researchers powerful tools to quickly and accurately process and interpret large volumes of speech data.

References

R. Baly, G.Da San Martino, J. Glass, and P.Nakov. 2020. We can detect your bias: Predicting the political ideology of news articles. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resni. 2014. Political ideology detection using recursive neural networks. Baltimore, Maryland, USA. Association for Computational Linguistics.

V. Kulkarni, J. Ye, S. Skiena, and W. Yang Wang. 2018. Multi-view models for political ideology detection of news articles. Association for Computational Linguistics.

ParlaMint. 2021. A multilingual comparable corpora of parliamentary debates.

Podium. 2020. Podium: Glove vectorizer. https://takelab.fer.hr/podium/package_reference/vectorizers.html?highlight=glovee.