Predicting French Parliament Member Affiliations Using NLP : A Deep Learning Approach Based on Parliamentary Statements

Final Project Report

Jérémie Stym-Popper
MVA ENS de Paris-Saclay
jeremie.stym-popper@ensae.fr

Gabriel Watkinson
MVA ENS de Paris-Saclay
gabriel.watkinson@ensae.fr

Abstract

In this report, we propose a deep learning approach for automatic recognition of French parliament member affiliations based on their declarations. Our study focuses on the field of Natural Language Processing (NLP), with the aim of training a neural network to accurately predict the political party affiliations of the deputies in the French Assemblée Nationale. To achieve this, we utilize data sourced from assemblee-nationale.fr and nosdeputes.fr, both of which provide systematic transcription of parliamentary statements. In addition, we leverage topic modelling techniques to extract the themes and subjects of the allocations in order to automatically provide a description of the deputies across the main topics discussed at the Assembly. To achieve accurate predictions, transformer-based algorithms are utilized, providing robust representations of French vocabulary. A classification neural network is trained to accurately identify party affiliations and associate them with relevant topics.

1. Introduction

What are some ways to determine a French National Assembly deputy's political orientation based on their speeches ? How can we identify the topics that a political party favours ? When are deputies more likely to speak up or remain silent ? These are all questions that can be answered using NLP methods. The objective is to understand the biases inherent in political interpretations based on speech and themes in a clear and precise manner. The study of the French National Assembly is particularly interesting for a few reasons. First, language is crucial in expressing political opinions and constructing politically-oriented discourse. Therefore, we utilized web scraping to gather transcripts of all assembly sessions from the official website. Second, the results may have public interest because they can help determine a deputy's political affiliation or shed light on themes and topics that are more frequently discussed by certain political groups. In other words, NLP analysis can provide valuable insights when applied to the speeches of French National Assembly deputies. This analysis can help us understand how certain topics can divide a given society and in what ways.

In addition, the use of Natural Language Processing (NLP) is a compelling approach because it can help to overcome certain obstacles that other methods like surveys and regression face. Surveys, for example, have several potential sources of bias that can influence the accuracy of the results. The questionnaire used in a survey may introduce *question bias* that can influence how participants answer. This can further lead to *response bias*, as some individuals may be less likely to respond to a survey than others, creating selection bias. Declarative bias can also distort the results of a survey. Additionally, surveys may have *measurement errors* because the choices of questions do not necessarily represent all the possibilities, and certain opinions may be forgotten because they were not considered at the beginning. Unlike surveys, an NLP model can capture the intentions of the participants, their motivations, and their personality, and it is less susceptible to these biases. While deep learning models have their own limitations such as generalization/overfitting and long and costly computations, they can still be a valuable tool for social and political sciences.

The report is organized as follows. First, we describe the related work both in the field of political analysis with NLP, and NLP models. Next, we define the problems we aim to solve, which involve a classification task and a topic modelling task. We then discuss our methodology, including data acquisition, processing, and visualization, as well as the different expe-

riments we conducted for both tasks. Our results, including quantitative metrics and a short quantitative analysis using both models, are then presented. Finally, we discuss the choices and limitations of our project.

2. Related Work

There is a rich body of literature exploring the intersection of NLP with political and social sciences, as evidenced by recent work by Ahmed [2022], Terechshenko et al. [2020] and Zhao et al. [2021]. Traditional methods in these fields have relied on survey data and statistical techniques such as regression analysis, but recent advances in deep learning have opened up new avenues for research. However, recent advancements in deep learning have provided new opportunities for research, even though these models can be computationally expensive to train. To mitigate this issue, transfer learning can be applied to word embedding techniques, which encode the semantic meaning of words and sentences as vectors, enabling classical classification methods. Transformers like BERT [4] and a French equivalent, CamemBERT [5] have become standard models for NLP, and we used these as a basis for our own model, which was designed to extract relevant features from the transcribed speeches of French deputies in the National Assembly. We also experimented with other transformer models, such as multilingual XLM-RoBERTa [6], as well as distilled versions of these models (see Section 4.2.2), in order to find the best model for our particular task.

To solve our classification problem, we took inspiration from Terechshenko et al. [2], who used transfer learning for political classification in the United States, comparing RoBERTa and XLNet for English text classification. Though, we use French language models.

Then, to get an idea of the topics mentioned in each intervention, Zhao et al. [3] propose a method that clusters the embeddings of politicians' speeches to gain an understanding of the topics covered in each intervention. We have been inspired by this method and have used BERTopic [7], which offers robust topic modelling through embeddings, dimension reduction, clustering, and weighting the words with a c-TF-IDF. From these topics, we have conducted qualitative analysis to assess the performance of our model on new data and to investigate if the topics influence the predictions.

3. Problem Definition

In this project, we aim to tackle two interconnected problems. Firstly, we focus on a classification task, where the objective is to predict the political views of French National Assembly deputies by analysing the content of their speeches. Our goal is to distinguish between left-leaning and right-leaning opinions. The second problem we aim to address is the extraction of topics from the interventions, which will enable us to analyse the deputies' tendencies.

3.1. Definition of the Classification Task

Initially, we attempted to predict the party affiliation of deputies, but found this task to be too complex due to the high degree of variation within and across political parties, and the fact that those parties change regularly, during the legislatures and totally at the start of a new legislature (every 5 year). We then attempted a three-classes classification (left-wing, center, and right-wing), based on our opinion, but found that the distinction between center and other parties was unclear. Therefore, we focused on the binary classification task, between the left and the right.

To solve this problem, we used pre-trained language models to generate features from the interventions and some additional context such as the title and theme of the session, and fed these features to another classification network that takes in a batch of embedded sentences and returns the predicted probabilities. This method is similar to the one proposed in Terechshenko et al. [2020].

We dispose of two highly different datasets, one for the 14th legislature (2012-2017), and one incomplete one for the 15th legislature (2017-2022), noted $\mathcal{D}14$ and $\mathcal{D}15$. Those datasets were built by scrapping nosdeputes.fr, and as we will see in the section about the data processing (Section 4.1.3), there are some issues with the data from the 15th legislature, therefore, we focus on $\mathcal{D}_{14}$.

$$\mathcal{D}_{14} = \{(I_i, T_i, C_i, P_i, f_i), (G_i, Y_i)\}_{i=1,\dots,n}$$

where I_i denotes the intervention, T_i denotes the title of the session in which the intervention was delivered, C_i denotes the context of the session (which is often empty), and P_i denotes the profession of the deputy speaking. The additional features f_i describe the deputy (we kept the age, gender, and number of previous mandates). It is important to note that I_i, T_i, C_i, P_i are all text data that need to be encoded in features to be used. Then, G_i denotes the political party of the deputy, and Y_i is the final label we want to predict, build from the political group G_i. It takes a value in either $\{0, 1, 2\}$ when doing the left-center-right classification, and $\{0, 1\}$ when doing the left-right classification. We build the maps $g : G \mapsto Y$ ourselves based on popular opinion regarding the parties. It is not that reliable, which is a reason why we drop the center parties. We

suppose that those datasets follow an underlying distribution noted p_{14} and p_{15}.

The problem we want to resolve is to find a way to build good features for the text input, among a list of possible language models $\mathcal{E} = \{h_{\text{BERT-ML}}, h_{\text{CamemBERT}}, h_{\text{RoBERTa-ML}}, \dots\}$, where h is a function that maps a sequence of words into a single vector. To do so, all the selected pre-trained language models provide a tokenizer, then a transformer to build contextualized embeddings for each token, which are then pooled (with mean pooling, CLS token, etc.) to obtain a single vector representing the entire sentence. We use these language models as feature-providing functions, and we will not fine-tune their weights. We then build a neural network classifier, which we will denote as f_θ, where θ represents the learnable parameters of the classifier. With these notations, we aim to find the classifier that minimizes :

$$\arg\min_{\substack{h\in\mathcal{E}\\ f_\theta}} \mathbb{E}_{\mathcal{D}_{14}\sim p_{14}}\left[\mathcal{L}(f_\theta(X_i), Y_i)\right] \tag{1}$$

where $X_i = (h(I), h(T), h(C), h(P), f)$, is the encoded input and $\mathcal{L}$ is the cross-entropy loss function.

We split our dataset in three, 50% to the train dataset, 20% for a validation dataset used to monitor overfitting during training, and 30% used for posteriori testing, the metrics we will show are measured on this test set. We approximate this expectancy with an empirical sum on a train dataset.

The task is challenging because of the presence of noisy data, such as short interventions without political content, it is a really hard task to humans as well, so we do not expect perfect accuracy.

3.2. Definition of the Topic Modelling Task

The second problem we address in this project is topic modelling. The main purpose of topic modelling is to build a probabilistic model to identify the themes and topics of a text. Here, the aim is to identify the main themes that we find in the speeches of the deputies at the National Assembly. This enables us to explore, first, whether the link between the subject on the agenda and the recurring themes is strong, and second, to link the main themes and subjects to a political trend. For example, left-wing parties will be more likely to talk about wage increases and tax increases. In Zhao et al. [2021], a similar idea was used to analyse the political landscape using clustering on embeddings.

However, we opt to use BERTopic Grootendorst [2022], which builds upon this approach. They demonstrate the feasibility of using a class-based TF-IDF procedure to cluster words and identify topics. The methodology follows the steps :

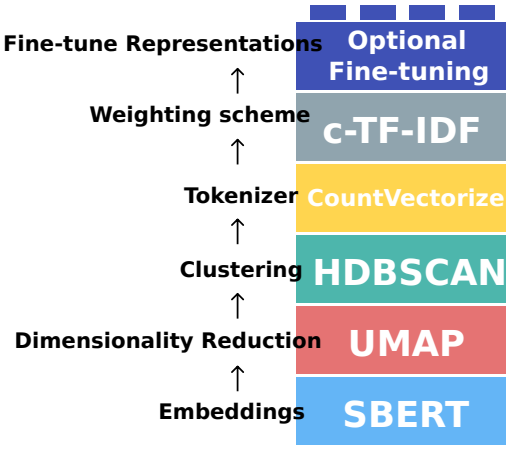


FIGURE 1 – Simplified overview of the function of BERTopic [1]

— First, the sentence embeddings are build with a pre-trained language model. the authors use the Sentence-BERT (SBERT) framework, but any embedding technique from the previously defined class $\mathcal{E}$ can be used.

— Second, a dimensionality reduction technique is used to optimize the clustering algorithm, as high-dimensional data can be difficult to handle. The authors propose using UMAP, a nonlinear dimensionality reduction technique that preserves local and global structure when reducing dimensionality. This structure is important, as it contains the information necessary to create clusters of semantically similar documents.

— After reducing the embeddings, we use a density-based clustering technique, HDBSCAN, to cluster our data. HDBSCAN can find clusters of different shapes and has the nice feature of identifying outliers where possible. As a result, we do not force documents into a cluster where they might not belong. This will improve the resulting topic representation as there is less noise to draw from.

— We then combine all documents of a cluster into a single document to represent the cluster. We count how often each word appears in each cluster to generate a bag-of-words representation. Finally, we use a modified class-based TF-IDF to rank the importance of the words inside a cluster, generating the topics we use.

$$W_{t,c} = tf_{t,c} \times \log\left(1 + \frac{\text{card}\, A}{f_t}\right)$$

where $W_{t,c}$ is the weight of the term t in the class c, $tf_{t,c}$ is the term frequency for the term t and

in class c, f_t is the term frequency of t across all classes, and A is the average number of words per class. The inverse document frequency is replaced by the inverse class frequency.

This approach is highly customizable, as each step has multiple option, but we chose to use the default values, except for the sentence embeddings, since they worked well on our data.

4. Methodology

In this section, we will talk about the decisions we made and our methodology.

4.1. Getting the Data

Our project is based on the rich and freely available textual data from the website of the French Assemblée Nationale. Indeed, the parliament's debates and decisions are transcribed by hand and published on the internet for citizens to access, providing us with a valuable source of information to analyse and draw insights from. We believe that this data is an excellent foundation for our project, and we are excited to use it to explore political opinions and topics discussed in the parliament. The availability of such data underscores the importance of transparency in political processes and the power of open data in promoting accountability and informed citizen participation.

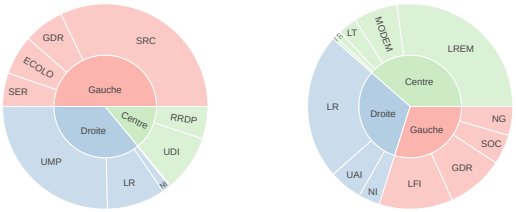
4.1.1 Data Scraping

To collect the data necessary for our project, we first attempted to access the official website of the French National Assembly [2], where the parliamentary transcripts are stored. However, we quickly discovered that extracting and processing the data from these XML pages would be a difficult and time-consuming task.

In search of a more efficient solution, we turned to the website nosdeputes.fr, which provided a more user-friendly API and had already done the web scraping for us. By using this API, we were able to gather over 750,000 interventions from more than a thousand deputies for both the 14th and 15th legislatures.

Although we only had access to a fraction of the data for the 15th legislature, we opted to use the entire dataset from the 14th legislature for training our models, with plans to conduct qualitative analysis on the 15th legislature. In our opinion, the decision to use an alternative website for data collection was a wise choice, as it allowed us to focus on the analysis rather than struggling with data extraction and formatting.

2. https://www.assemblee-nationale.fr/



(a) 14th legislature (2012-2017)

(b) 15th legislature (2017-2022)

FIGURE 2 – Distribution of the political groups

4.1.2 Data Cleaning and Preprocessing

After acquiring the data from the NosDéputés website API, we conducted some data cleaning and preprocessing. We observed HTML artifacts and encoding errors that were present in the data and removed many interventions that were purely organizational in nature, such as those made by the president of the assembly or short exclamations.

The resulting dataset contained the interventions, along with their context, including the session title, date of the session, and other information related to the deputies, such as their political affiliation, age, and profession.

For our modelling experiments, the primary features of interest were the textual data of the interventions and the textual data associated with the session title and context. We also considered features directly related to the deputies, their age, sex and textual description of their profession, but due to the small number of deputies in our dataset, using these features alone could result in overfitting, as we observed during our experiments.

4.1.3 Data Exploration

We present descriptive statistics for the data from the 14th legislature, and some for the 15th as well, albeit to a lesser extent.

Two pie charts are shown in Figure 2, displaying the distribution of political groups for both legislatures. The parties have undergone considerable change, and our classification of parties into Left, Center, and Right is really different between the two periods. The emergence of the LREM party contributes to this ambiguity, which led us to train our model on the 14th legislature, where the distinction between Left and Right was clearer. This way, we could use our model on the deputies from LREM to obtain an idea of their political affiliation.

	14th legislature	15th legislature
Nb of deputies	648	659
Left	309	72
Center	52	405
Right	287	155
Nb of professions	331	305
Men	472	391
Women	176	268
Nb of speeches	743,254	506,283
After processing	363,572	215,692

TABLE 1 – Summary of the collected data

Then, we looked at the distribution of the features of the deputies of the 14th legislature. The Appendix 1a shows the distribution of the gender at the parliament, grouped by political orientation. We see that for all three orientations, there are many more men than women, with 80% men and 20% women in both the Right and Center, and a slightly more progressive ratio of 65/35 for the Left.

Then, the Appendix 1b shows the distribution of the number of mandates already served by the deputies. We see that most of them have served 2 mandates, and the distribution across groups is very similar.

The Appendix 1c shows the distribution of the ages of the deputies. We see that they were all born after 1940 and the distributions are slightly different. The Right has older deputies, whereas the Left and Center are younger.

We observed the number of interventions by party and the intervention lengths in both legislatures (Appendix 2). Most of the deputies in both legislatures had 100 interventions or fewer, while some had over a thousand interventions.

Lastly, the Appendix 3 displays that the number of words in each intervention is mostly below 200 words, with only a few interventions longer than 500 words. This is important to consider because most pretrained language models have context memory of only a few hundred tokens, so we don't want interventions that are too long, or we would need to truncate them.

The results of the data collection and processing were summarized in Table 1.

4.2. Classification Task

The first objective we want to solve is the classification task. To do so, we have to select both a language model to provide embeddings for our textual inputs, and then we have to select a neural net classifier. We will explore different options for both subtasks.

4.2.1 Baseline Models

To establish a comparison point for our results, we implemented two baseline models.

Dummy Classifier : The first model is a dummy classifier, which does not learn anything and only makes predictions based on simple statistics from the training data labels. The "prior" strategy was used, which always predicts the most present class, and returns the empirical class distribution of Y for the predicted probabilities.

BERT Baseline : The second baseline model is a BERT multilingual model, which is more complex. This model relies on a neural network that takes as inputs the interventions, the title of the session, and the context of the session. These inputs are embedded into 768 dimensional vectors by taking the embedding of the CLS tokens of the `bert-base-multilingual-cased` model from HuggingFace. We then project these embeddings in the same space with a linear layer each, followed by a GELU activation function. These 3 vectors are then added and fed into a 2-layer perceptron, returning the 2 or 3 logits, depending on the problem we select (see Appendix 4). It is important to note that we only use the intervention and the title of the session. The reason for this is to have a model that can be used on new deputies or other speakers, on whom we do not have any information.

To optimize the classifier, we used the ADAM optimizer with default parameters and an initial learning rate of 0.001. We also used a learning rate scheduler that divides the learning rate by 10 if the validation loss does not improve after 3 epochs. Additionally, we performed early stopping if the validation loss did not improve after 15 epochs and kept the epoch with the lowest validation loss.

4.2.2 Selection of the language model

As discussed during the definition of the problem. We want to use pre-trained language models to extract features from our textual inputs, without fine-tuning their weights. Therefore, having a good initial model is important. We decided to compare the models presented in Table 2, with the same classification head and same experimental setup as the BERT baseline. The comparison is not exhaustive as we didn't fine-tune each model, and they could therefore perform better than what we evaluated. But we believe this is already an interesting benchmark.

	Nb params	Embedding dim	Pooling	Max seq length	Nb languages	Training dataset	Dataset size	Company trainer
bert-base-multilingual-cased	345M	768	CLS	512	104	Wikipedia	16GB	Google AI
camembert-base	110M	768	Mean	512	French	OSCAR	138GB	Facebook AI
xlm-roberta-base	550M	768	CLS	512	104	CommonCrawl	2.5TB	Facebook AI
paraphrase-multilingual-mpnet-base-v2	117M	768	Mean	128	50+	custom		Huggingface
paraphrase-multilingual-MiniLM-L12-v2	110M	384	Mean	384	50+	custom		Microsoft AI
distilcamembert-base		768	Mean	512	French	OSCAR		Facebook AI
distilbert-base-multilingual-cased	134M	768	Mean	512	104	Wikipedia	16GB	HuggingFace
distiluse-base-multilingual-cased-v2	134M	768	Mean	128	104	Wikipedia		HuggingFace
bert-tiny	14.5M	312	Mean	128	English	Wikipedia	16GB	Google AI

TABLE 2 – Summary of the pre-trained language models

4.2.3 Precomputing the Embeddings

Since we only use the pre-trained language models as feature builders, we can pre-compute the embeddings of the inputs once, then train the classifier from them, this significantly increases the speed of computation (from 1h per epoch on some models, to minutes).

4.2.4 Further improve the Base Classification

Once we selected our language model, we try to improve the performance of our model by choosing a better optimizer, learning rate and scheduler. We tested standard optimizers such as SGD, Adam, AdamW and Adagrad. We also tried different learning rate schedulers : ReduceLROnPlateau, Exponential, Cosine Annealing, ...

4.2.5 Testing other Architectures

We also perform experiences with larger networks, with more layers and neurons, and with concatenation of the inputs instead of addition. We then used the additional features on the deputies (profession, gender, age, ...).

Lastly, we tried other architectures : we used a self-attention head, that looks at the three inputs (the intervention, title and context) and contextualizes them appropriately, in a transformer style classification head, that has multiple layers with intermediate feed forward layers and layer norm. This is an interesting alternative to simply adding or concatenating the three textual inputs.

4.3. Topic Modelling

We used the data from the 15th legislature in this experiment. We focused more particularly on the parties LREM, LFI and NI. We kept the interventions that are neither too short nor too long (between 16 and 64 words).

For the sentence embedding method, we used the one we deemed the best in the previous task, which is a mean pooling CamemBERT-base (see the Section 5.1.3). We then used the default configuration of BERTopic, which is a UMAP with 5 components, 15 neighbours and using the cosine metric, and a HDBSCAN with 15 minimum cluster size and the Euclidean metric.

5. Results

The experiments were realized with PyTorch, using the PyTorch Lightning package to perform the training and evaluation. The training metrics were logged with Tensorboard. Lastly, the pre-trained models come from HuggingFace. Our entire code is available on our GitHub repository : https://github.com/gwatkinson/NLP_Assemblee. It also contains all the logs and results present in this report. However, the data was too voluminous to be hosted on GitHub or sent by mail.

5.1. Results of the Classification Task

5.1.1 The Dummy Classifier

First, we evaluated a Dummy Classifier in order to have the first baseline to which the models are compared. The dummy model using the uniform strategy gives us a log-loss of 0.680, a balanced accuracy of 0.5, a F1 score of 0.452 and an AUC of 0.5. This is expected as it always returns the same class. Those are values that all our classifiers should beat.

5.1.2 Precomputed Embeddings

Since we want to evaluate different language models, we find it interesting to look at the disposition of the embeddings. To do so, we focused mainly on the embeddings of the title, and the interventions. The Appendices 5 and 6 show examples for CamemBERT and RoBERTa. Each colour refers to a political trend : green is left-wing, red is right-wing, yellow is center. Overall, we notice that the sentences which belong to the same political trend are located next to each other. We also notice that the embeddings have significantly different structures between CamemBERT and RoBERTa.

	Nb Epoch	Nb params	Log Loss	Accuracy	Recall	Precision	F1-score	AUC
camembert-base	12	2.36M	**0.537**	**0.718**	**0.605**	0.684	**0.642**	**0.790**
paraphrase-multilingual-mpnet-base-v2	15	2.36M	0.553	0.706	0.517	**0.701**	0.595	0.778
xlm-roberta-base	27	2.36M	0.555	0.709	0.591	0.674	0.630	0.777
paraphrase-multilingual-MiniLM-L12-v2	17	1.48M	0.559	0.702	0.575	0.667	0.617	0.769
distilcamembert-base	17	2.36M	0.565	0.700	0.625	0.646	0.636	0.765
distilbert-base-multilingual-cased	18	2.36M	0.568	0.696	0.583	0.654	0.616	0.762
distiluse-base-multilingual-cased-v2	11	1.77M	0.571	0.695	0.579	0.652	0.613	0.758
bert-base-multilingual-cased (Baseline)	49	3.41M	0.583	0.683	0.544	0.643	0.589	0.744
bert-tiny	18	0.89M	0.604	0.668	0.584	0.607	0.596	0.719

TABLE 3 – Comparing metrics for different language models, using the same architecture as the baseline, the same Adam optimizer, a learning rate of 0.01 and a scheduler that monitors the validation loss.

5.1.3 Language Models Comparison

In this section, we look at the results of the experiments comparing the language models on the same task, with the same neural net and optimizer. We can see in the Table 3 that the best performing model is CamemBERT, which is an implementation of RoBERTa for the French language. This is what we expected, as single language model tend to perform better than the more complex multilingual models. Furthermore, the vocabulary used at the Assembly can be very specific, so we expect French language-models to perform better as they have more French training data. It achieves the best results across most metrics, with a loss of 0.537 and a balanced accuracy of 71.8%.

On another hand, the baseline bert-base-multilingual-cased performs poorly (worse than its distilled version) which is unexpected because the architecture of the BERT model is supposed to be more complex and have more parameters than its distilled version, and it has been trained on a larger corpus of data. The accuracy given by the bert-base-multilingual-cased is 0.683, which is smaller than distilbert-base which has an accuracy of 0.695.

We see that bert-tiny is still better than the random classifier, despite being a small English model. This was not expected because the data on which the model is trained are uniquely French text, so that the bert-tiny model is *a priori* unable to perform well. However, the bert-tiny model gives an accuracy of 0.604 whereas the dummy classifier presented before gives an accuracy of 0.498. So, even with the "wrong" language, the classifier performs better than randomness.

Those results are displayed in the Appendix 7 displaying the evolution of the losses during the training.

5.1.4 Choosing the Optimizer

From this model and classifier architecture, we tried to fine-tune the training procedure to see if we could improve our results. We thus tested different optimizer from the default Adam, different learning rates and schedulers. The Table 4 displays the results of our experiments. We chose to keep a selection of tests that we thought were significant, but others were conducted.

The results are quite varied. We do not see a single best optimizer. Default Adam with a learning rate of 0.001 and exponential scheduler achieves the best accuracy and AUC. But slightly tweaking the betas parameters improves the cross entropy loss and the precision, while keeping the same AUC. However, this seems to show that a learning rate of 0.001 and an exponential scheduler with a gamma of around 0.9 per epoch works well with Adam and AdamW. The Appendix 8 summarizes the evolution of the loss and accuracy during the training.

5.1.5 Results of the other architecture

We are now happy with our simple classifier that relies on a multi layer perceptron, it has a 7% bump in accuracy compared to our baseline. However, we didn't use all the features at our disposal. We indeed have information on the deputies (profession, gender, age), and the pooling layer for our three inputs could be improved. The Table 5 presents the results of our experiments.

We notice that the model **Features+Profession (3L@1024)** that uses the profession as an input, reaches an accuracy of almost 100%. This is due to the fact that the professions are almost unique to each deputy, therefore, the model can identify the deputies from their profession and then predict the correct class. We therefore drop this model and feature all together.

We see that only using the interventions works quite well, when using a bigger network. We obtain a loss of 0.555 and a F1 score of 0.61. This is better than the baseline. However, the title and context of the session and the additional features provide quite a lot of information.

Optimizer	LR	LR Scheduler	Last Epoch	Log loss	Accuracy	Recall	Precision	F1-score	AUC
$\text{Adam}_{\beta=(0.91,0.997)}$	0.001	ExponentialLR0.93	19	**0.489**	0.752	0.654	**0.727**	0.689	**0.834**
Adam	0.001	ExponentialLR@0.90	25	0.490	**0.753**	0.669	0.721	0.694	**0.834**
$\text{AdamW}_{AMSGRAD,WD=0.01}$	0.001	ExponentialLR@0.92	19	0.491	**0.753**	0.674	0.718	0.695	**0.834**
$\text{AdamW}_{\beta=(0.85,0.997),WD=0.005}$	0.001	ReduceLROnPlateau@0.10	14	0.493	0.752	**0.681**	0.713	**0.696**	0.833
AdamW	0.001	ExponentialLR@0.99	15	0.494	0.751	0.640	0.732	0.683	0.833
Adam	0.001	ReduceLROnPlateau@0.10	13	0.494	0.749	0.677	0.710	0.693	0.831
$\text{AdamW}_{\beta=(0.99,0.999),WD=0.005}$	0.0001	ReduceLROnPlateau@0.10	49	0.511	0.735	0.630	0.706	0.665	0.814
Adagrad	0.001	ReduceLROnPlateau@0.10	48	0.558	0.705	0.579	0.672	0.622	0.772
Adagrad	0.001	ExponentialLR@0.75	26	0.581	0.691	0.552	0.656	0.600	0.750
SGD	0.01	ReduceLROnPlateau@0.50	49	0.596	0.682	0.512	0.653	0.574	0.734

TABLE 4 – Results of our experiments to choose the optimizers, on the test dataset. Only sensible models were kept, a lot of other test were not displayed to keep things simple.

	Nb Epochs	Nb params	Log Loss	Accuracy	Recall	Precision	F1-score	AUC
Features+Profession (3L@1024)	19	5.25M	0.096	0.961	0.952	0.955	0.954	0.994
Transformer Head (4H+3L@384)	49	4.60M	**0.462**	**0.771**	**0.723**	0.728	**0.726**	**0.856**
Transformer Head (4H+2L@256)	14	1.72M	0.467	0.767	0.681	0.741	0.710	0.852
Transformer Head (8H+2L@256)	21	1.72M	0.475	0.764	0.669	**0.742**	0.704	0.849
Features+Concat (3L@384)	14	3.12M	0.491	0.750	0.662	0.720	0.689	0.831
Standard (3L@1024)	14	8.74M	0.502	0.748	0.699	0.699	0.699	0.831
Only Intervenions (4L@1024)	15	3.94M	0.555	0.706	0.546	0.687	0.609	0.775

TABLE 5 – Results of our experiments to test the different architectures, on the test dataset. For the MLP, the notation *4L@1024* signifies that there are 4 layers with 1024 neurons. For the transformer like models, we added the number of heads to the notation.

On another hand, we see that adding layers and neurons do not increase the performance of the model significantly. Using concatenation instead of addition, yields the same results.

Lastly, the transformer like architecture seems to perform the best, in all three configuration. The one with the most parameters (**Transformer Head (4H+3L@384)**) and epoch is the best. This is interesting because it does not have many more parameters (due to a lower hidden dimension size) than the standard experiments.

To resume all those results, the Appendix 9 shows the evolution of the loss and accuracy of the different models during training.

5.1.6 Conclusion on the Classification Task

In conclusion, the dummy classifier gave us a log-loss of **0.680**. We managed to improve it with the baseline model, bringing it to **0.583**. Selecting the correct language model further improves the loss to **0.537**. Additional experiments bring this loss to **0.489** by selecting the correct optimizer and hyperparameters. Lastly, changing the pooling method to a Transformer-like mechanism improve the loss to **0.462**. This is the model we decided to keep, as it performs the best in most of the metrics.

5.2. Results of the Topic Modelling

Using topic modelling on the data from the 15th legislature, we obtained clusters of interventions. The Appendix 4.3 presents the most important words in each topic. For instance, the main topic discussed at the Assembly is about what is related to tax, retirement and Euro, so topics about economics. These topics depend on what is on the agenda, and therefore depend on the deputies who are actually present at the Assembly.

We can also note the presence of clusters about democracy and referendum (topic 5), energy and nuclear industry, the medical sector and hospitals, the European Union, or events related to current news, such as the Yellow jacket movement, or the Benalla scandal. This is interesting because we expect our model to perform better on certain topic that are more divisive, such as taxes or ecology, and worse on topics less polemical. For instance, we expect that a topic like ecology could better reveal the political position of a deputy, since the left-wing parties traditionally advocate for these causes.

The Appendix 12 presents some of the main topics in the reduced embedding space. We see that similar themes are close to each other. While topics related to particular events are on the extremities as they are

quite unique.

5.3. Qualitative analysis

The topic modelling coupled to the classification model allows us to see what position a member of parliament holds on certain topics. This allows us to classify them on the left or on the right according to his opinion on the subject (note that his position can vary according to the subject, regardless of his original political affiliation).

In this section, we will look at some examples bringing both models together. We use topic modelling to estimate the probability of being left-wing. For instance, in the Appendix 14, we can see that a left-wing deputy is indeed categorized on the left on most of the topics. About the Yellow jackets or taxes, his position is left-wing. However, for other topics like "general interest" and "amendment", there is more uncertainty, or even misleading classifications. The Appendices 13, 16 and 15 present the same example on other deputies.

6. Limits and Discussions

Even if the benchmark is dense and offers plenty of comparisons with other models, there are still some limitations to our way of doing things, and we can always discuss the relevance of our results.

6.1. Performance

We notice that the classification is not always accurate when it comes to the topic modelling. Indeed, we can see on Appendix 13 that on most of the presented topics, a deputy normally classified on the right like Marine Le Pen is rather classified on the left, even on topics about immigration ! This proves that our model can be wrong in extreme cases. This can also be emphasized by the fact that the data for the topic modelling and classification are from different legislatures. This still means that our model is limited.

6.2. Data quality

One of the first issue we had to cope with was the discontinuity between legislatures. Indeed, the political map of the assembly changes every 5 years, which makes our classifier sensible to out of distribution data (with new deputy for instance). This limits the features we can use. We saw in Section 5.3, that using the classifier on the 14th legislature does not provide great predictions on deputies from the 15th. Lastly, the preprocessing done by the French Assembly also changes between legislation, and some information is lost (or gained) in this process.

6.3. Feature selection

In our final model, we only use camembert-base as feature provider, which limits our scope, we could have used a combination of different embedding providers to improve our model (CamemBERT in parallel to FastText, xlm-roberta, ...). Furthermore, we did not fine-tune the embedding model because it was too computationnaly expensive. Since the language spoken at the French National Assembly can be very particular, we expect that fine-tuning the embedding model would bring benefits. Finally, we did not use camembert-large because, for similar reasons, it can be very expensive to train and to use, so it limited the number of words and parameters we use in our model. Lastly, in a real world situation, it could be beneficial to use distilled versions of BERT or CamemBERT, since they have a smaller footprint and are faster. Even though, they are slightly less efficient, as shown in the previous section.

6.4. Exhaustiveness

Our experiments are not exhaustive, many options and combinations of hyperparameters were not tested. For instance, instead of training a classification model, we could have trained a regression where the results stand between -1 (left-wing) and 1 (right-wing). This could give a more accurate precision and can locate a deputy more or less near to the center (and it gives an idea of its political party, no just its political trend).

6.5. Overfitting

There are not enough deputies to train perfectly the model. We should avoid relying on personal features from them. Since relying on features like the deputy's job could cause a data-leakage effect during the prediction phase, where we can identify the deputies. Avoiding this is better since it makes the model more general (accepting new or unknown speakers).

6.6. Difference with the Poster

Since the *poster session*, we have decided to change two crucial points in our approach. First, because of problems in our data distribution, we switched to a binary classifier (left-wing, right-wing). Moreover, as mentioned above, the profession of the deputy caused some issues, so we replaced this input with more "context" about the session (agenda, date, etc.). This improves the performances of the model.

6.7. Extension

We used two models on two different legislatures : the 14th to train the classifier and the 15th for topic modelling. We have seen that the two models combined

give interesting results, because they allow us to find the topics mainly addressed by one political party or the other. It also allows highlighting some failures of the model, in particular with the classification of some MPs in the wrong political side for most of the topics discussed.

An idea to go further would be to use the results of the topic modelling to refine those of the binary classification. For example, we could highlight words that are particularly divisive, and that classify the right and left deputies well. By giving more importance to these words in the classification, one could assume that the results would be better.

7. Conclusion

In this study, we aimed to predict the political affiliation of the deputies in the French National Assembly based on their statements and the topics discussed during parliamentary sessions. We utilized embeddings provided by pre-trained classifier (CamemBERT, BERT, ...) to build a classification model that considered all the inputs. We incorporated a topic modelling component to predict the left-wing or right-wing trends of various topics and to redefine the position of a deputy on a particular topic. After comparing several pre-trained language models, we found that the CamemBERT model was the best for the National Assembly's specific vocabulary. We also experimented with different optimizers and model architectures to determine the best construction and hyperparameters.

While there are some limitations to our model, we achieved good metrics and found intriguing results. We observed that it is possible to predict a deputy's political trend based on their statements and positions on various themes and topics. We conclude that our study has the potential to be useful in political analysis and policy-making. Further improvements can be made by addressing the grey areas that remain and refining the model's accuracy.

Références

[1] Nabib Ahmed. *Natural Language Processing Techniques for Political Opinion and Sentiment*. PhD thesis, Harvard University, 2022. 2

[2] Zhanna Terechshenko, Fridolin Linder, Vishakh Padmakumar, Michael Liu, Jonathan Nagler, Joshua A Tucker, and Richard Bonneau. A comparison of methods in political science text classification : Transfer learning language models for politics. *Available at SSRN 3724644*, 2020. 2

[3] He Zhao, Dinh Phung, Viet Huynh, Yuan Jin, Lan Du, and Wray Buntine. Topic modelling meets deep neural networks : A survey, 2021. URL https://arxiv.org/abs/2103.00498. 2, 3

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805. 2

[5] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. Camembert : a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. 2

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta : A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692. 2

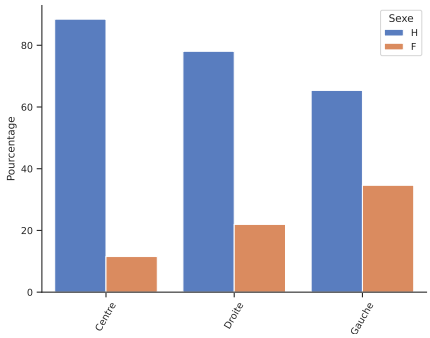
[7] Maarten Grootendorst. Bertopic : Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv :2203.05794*, 2022. URL https://arxiv.org/abs/2203.05794. 2, 3

[8] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116, 2019. URL http://arxiv.org/abs/1911.02116.

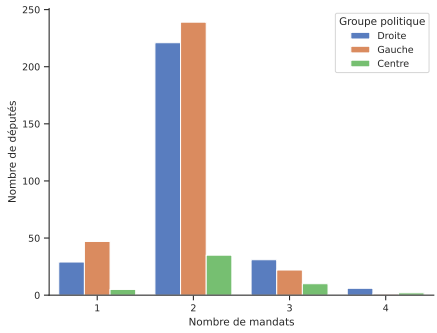
[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip : Compressing text classification models, 2016. URL https://arxiv.org/abs/1612.03651.

Appendices

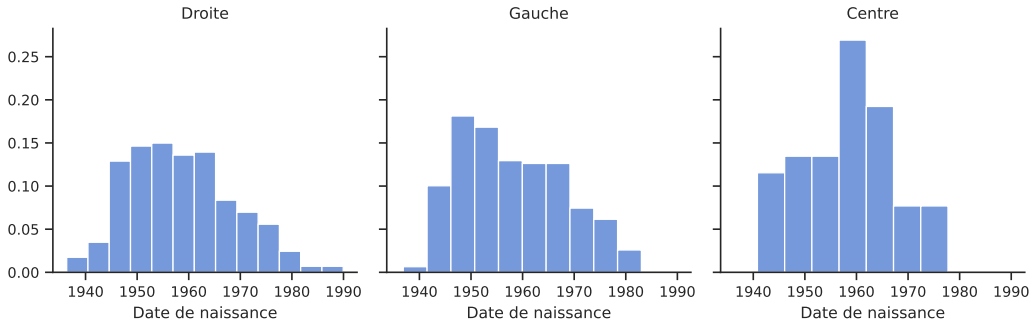
A. Description of the dataset



(a) Distribution of gender by political orientation

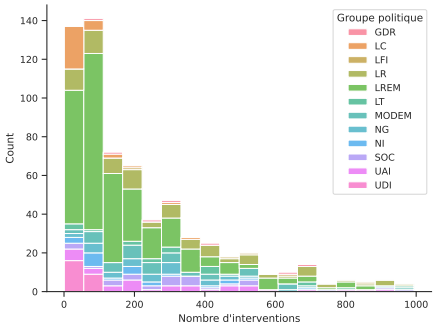


(b) Distribution of the number of mandates

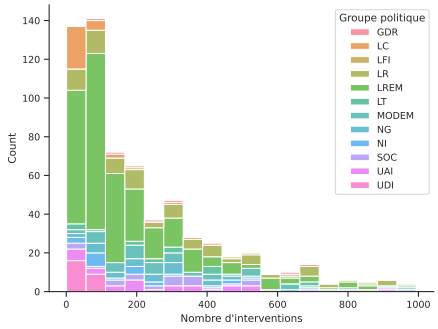


(c) Distribution of age by political orientation

Appendix 1 – Distribution of features of the deputies of the 14th legislature

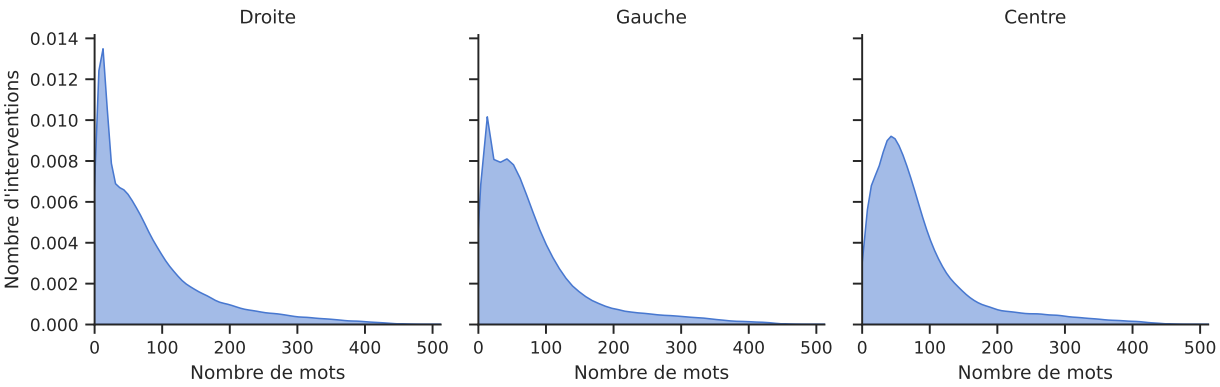


(a) Number of intervention by party for the 14th legislature



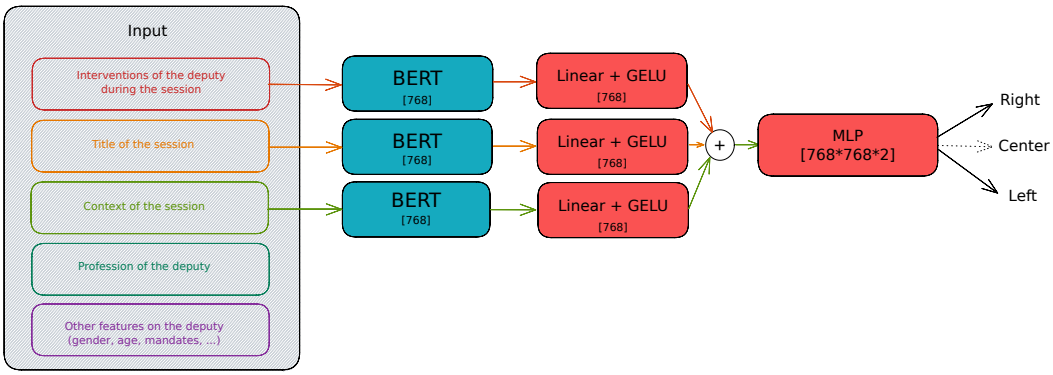
(b) Number of intervention by party for the 15th legislature

Appendix 2 – Distribution of the number of interventions by party



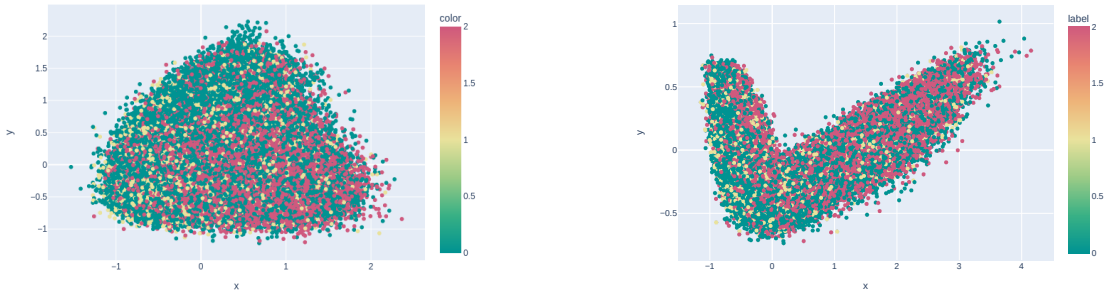
Appendix 3 – KDE of the number of words in each intervention

B. Additional graphs of the Experiments



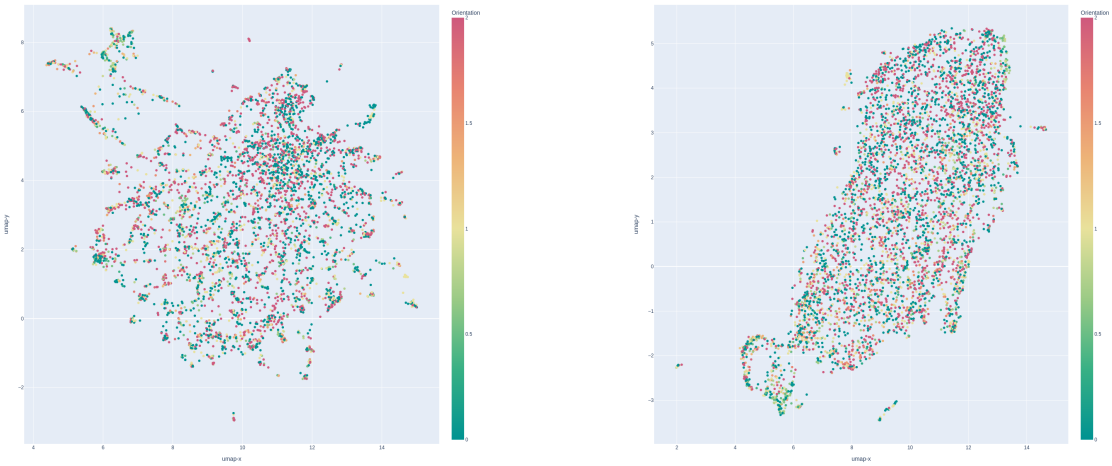
Appendix 4 – Schema of the baseline architecture

B.1. Visualization of the embeddings



(a) CamemBERT embeddings for the interventions (b) RoBERTa embeddings for the interventions

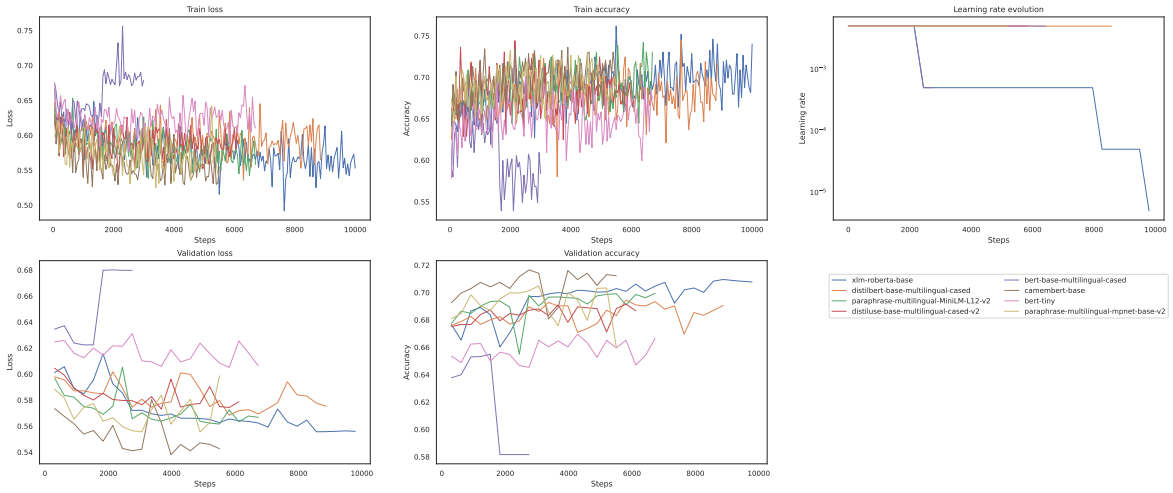
Appendix 5 – Visualization of the embeddings of the interventions, for CamemBERT and XLM-RoBERTa multilingual (left : green, center : yellow, red : right)



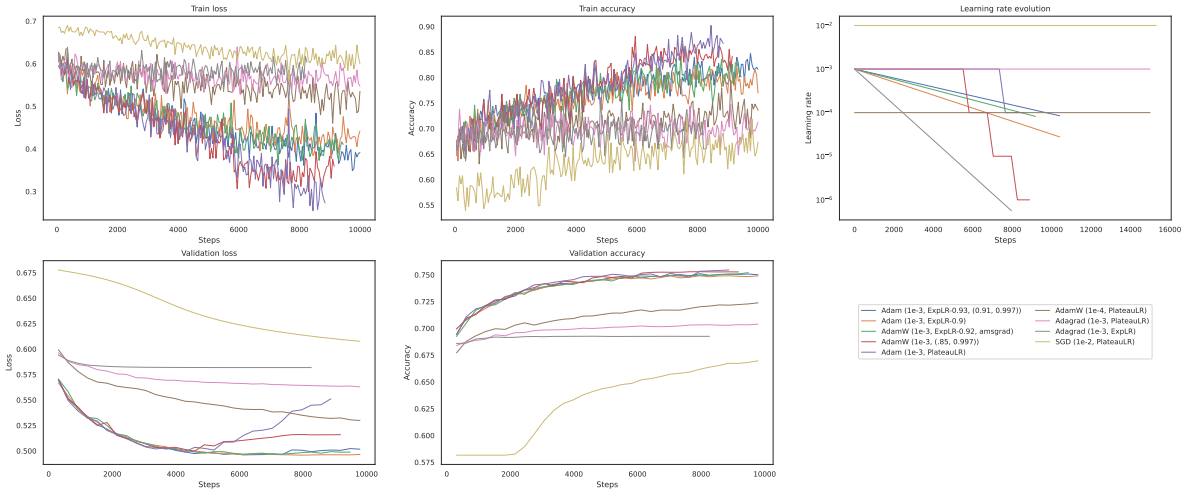
(a) CamemBERT embeddings for the titles (b) RoBERTa embeddings for the titles

Appendix 6 – Visualization of the embeddings of the titles, for CamemBERT and XLM-RoBERTa multilingual (left : green, center : yellow, red : right)

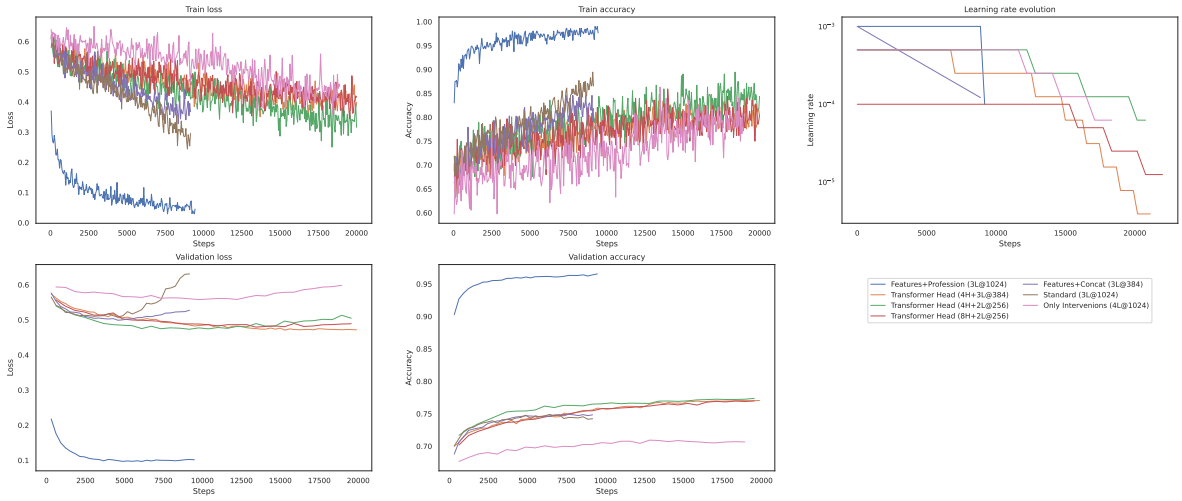
B.2. Evolution of the losses during the experiments



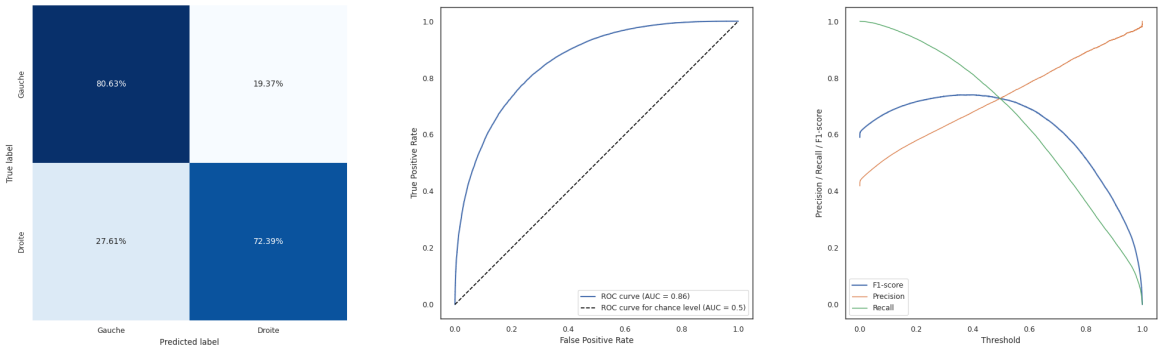
Appendix 7 – Evolution of the train and validation loss and accuracy during the training of the experiments comparing the language models.



Appendix 8 – Evolution of the train and validation loss and accuracy during the training of the experiments on the optimizers, and the evolution of the learning rate value.



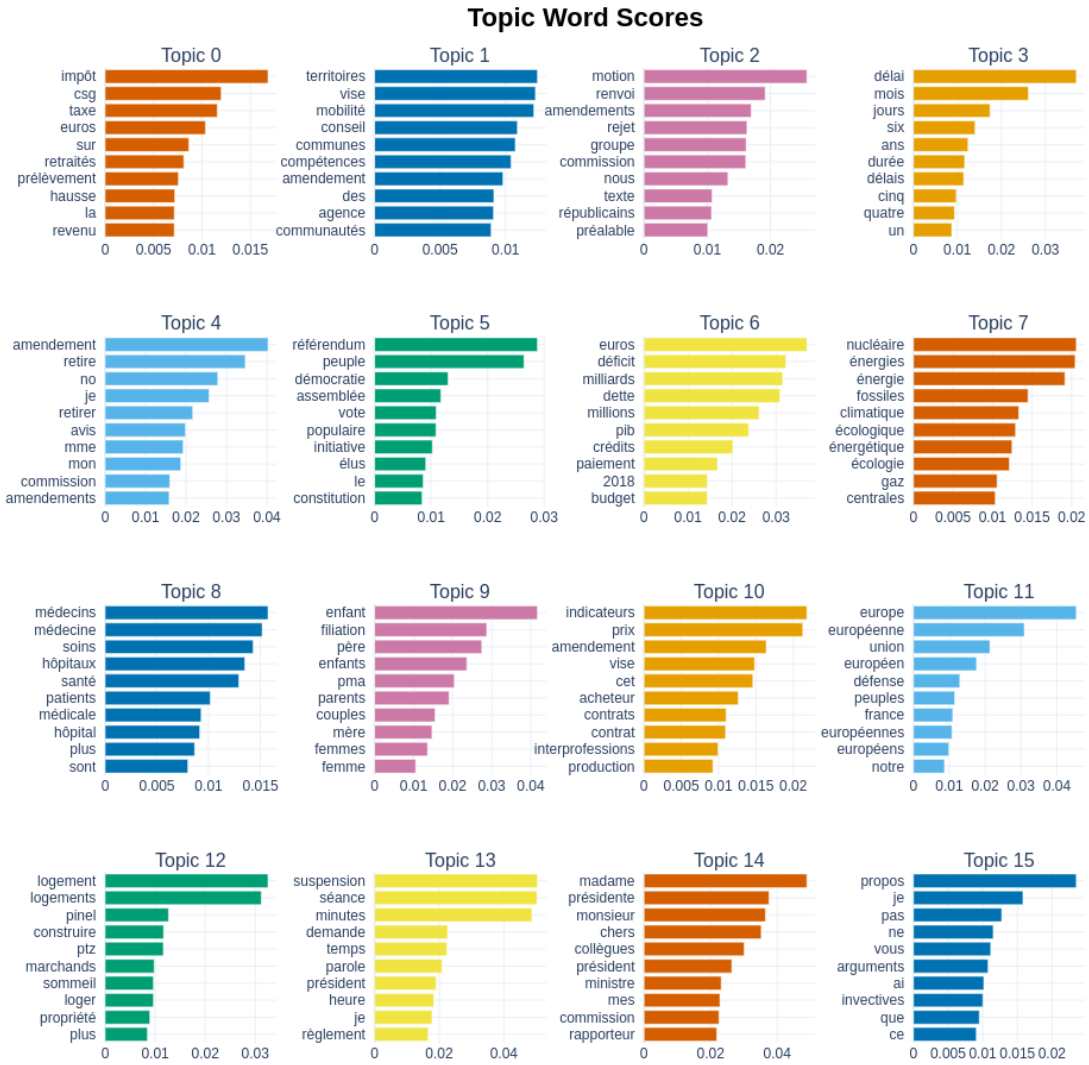
Appendix 9 – Evolution of the train and validation loss and accuracy during the training of the different architectures, and the evolution of the learning rate value.



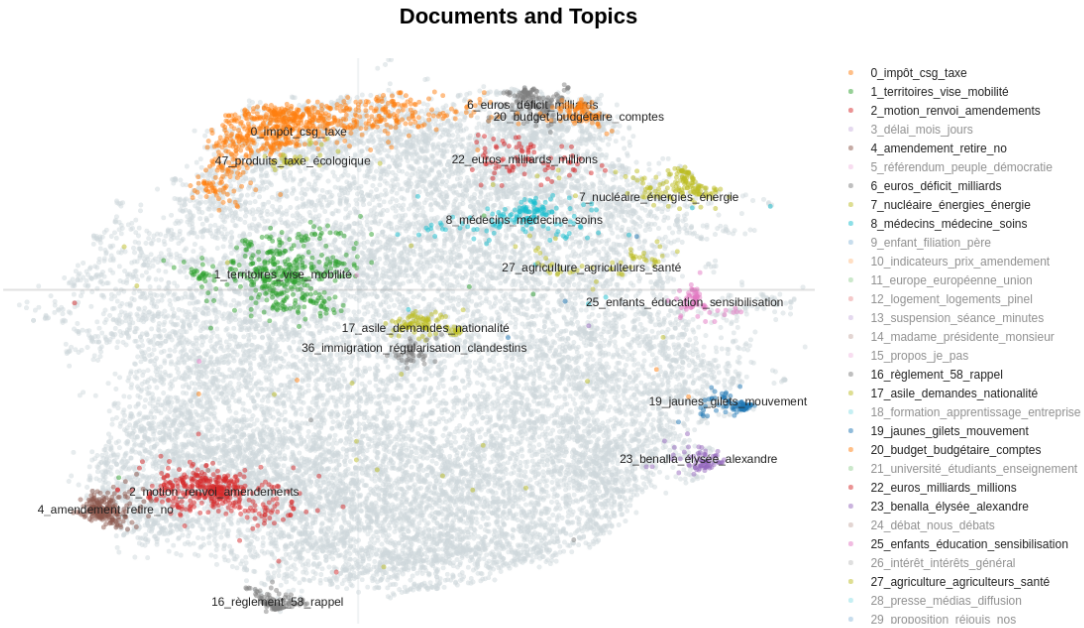
(a) Confusion matrix (b) ROC curve (c) Precision-Recall curve

Appendix 10 – Metrics and graphs of the best model (**Transformer Head (4H+3L@384)**)

B.3. Results of the Topic Modelling

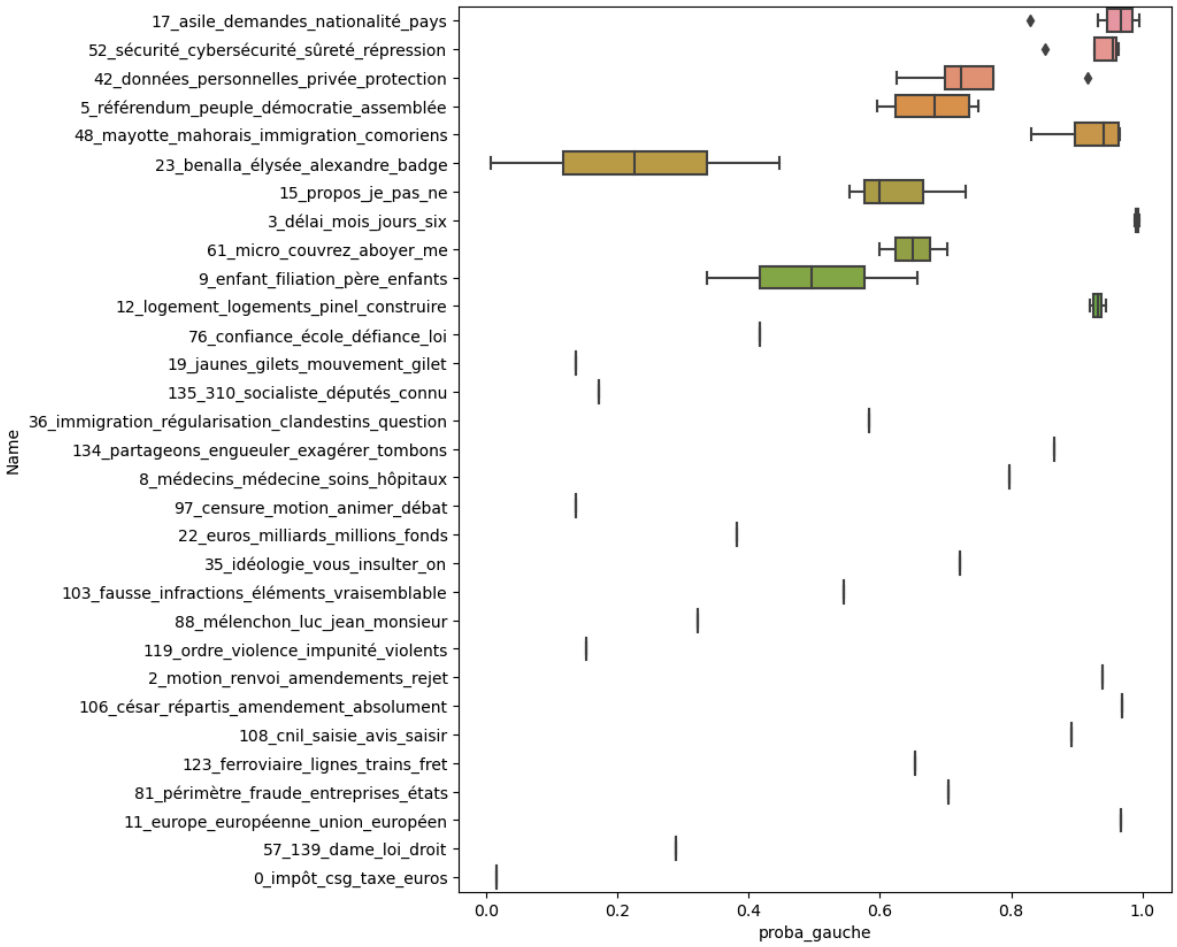


Appendix 11 – Top 16 cluster drawn from the topic modelling experiments

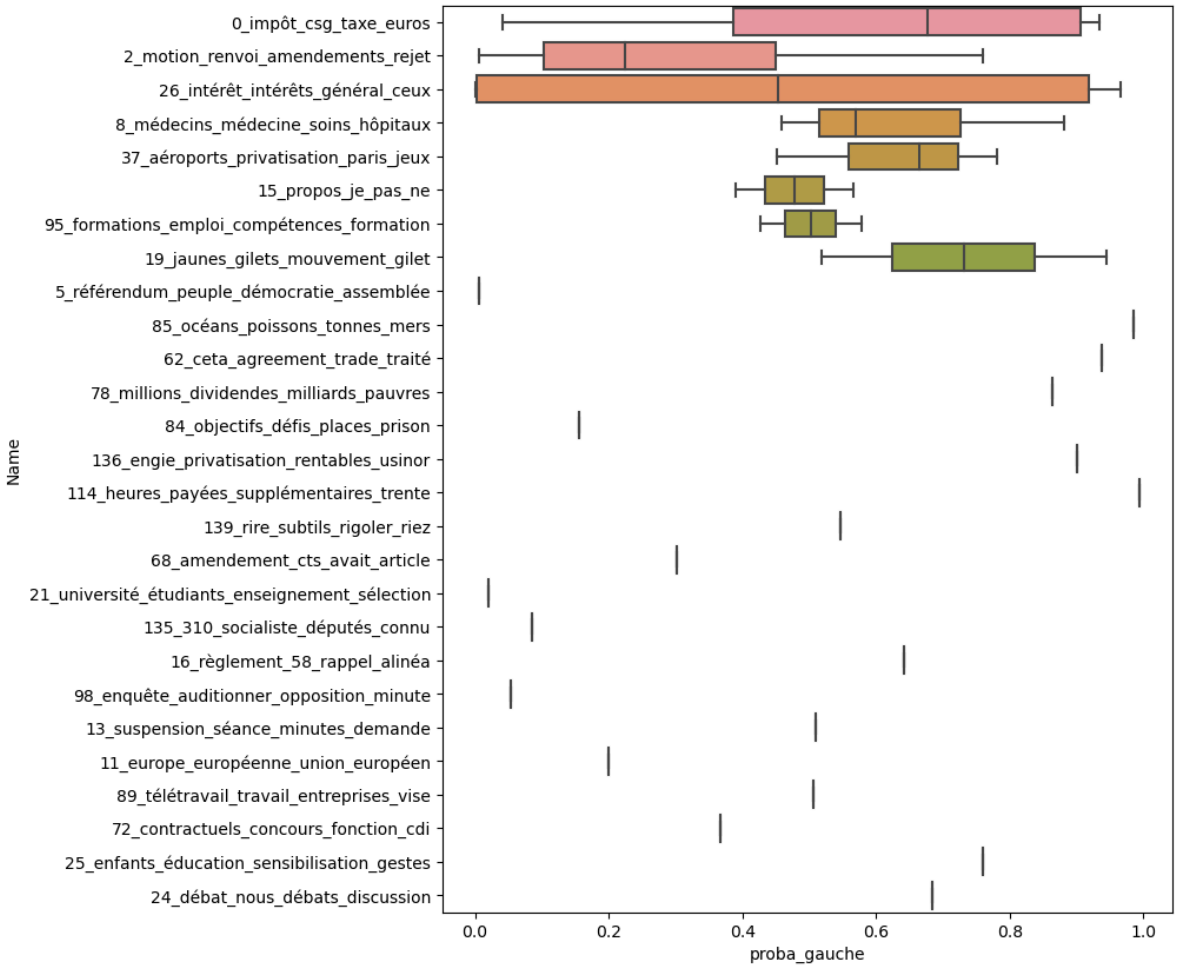


Appendix 12 – Scatter plots of the Embeddings of the interventions, with some interesting topics highlited

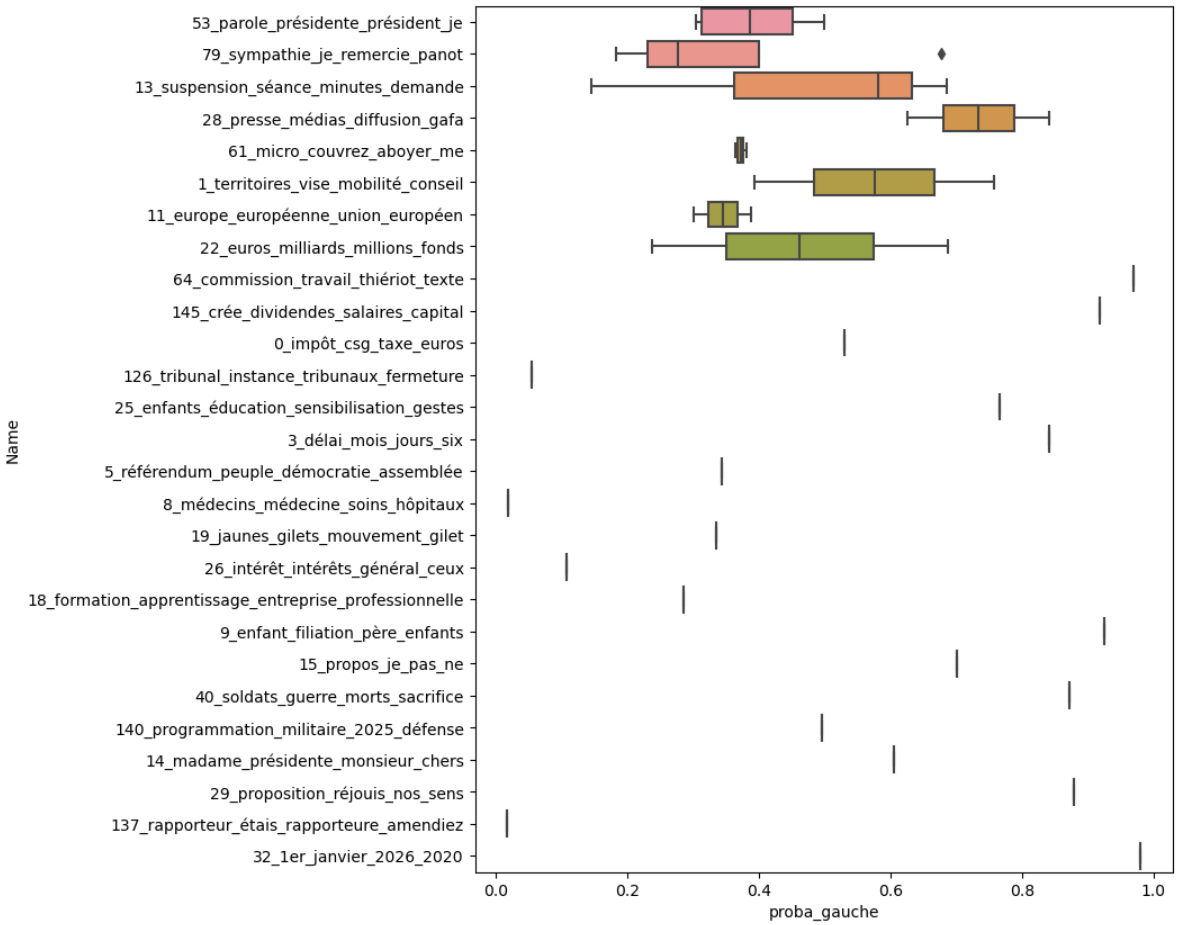
C. Cross Analysis using both the Classifier and Topic Model



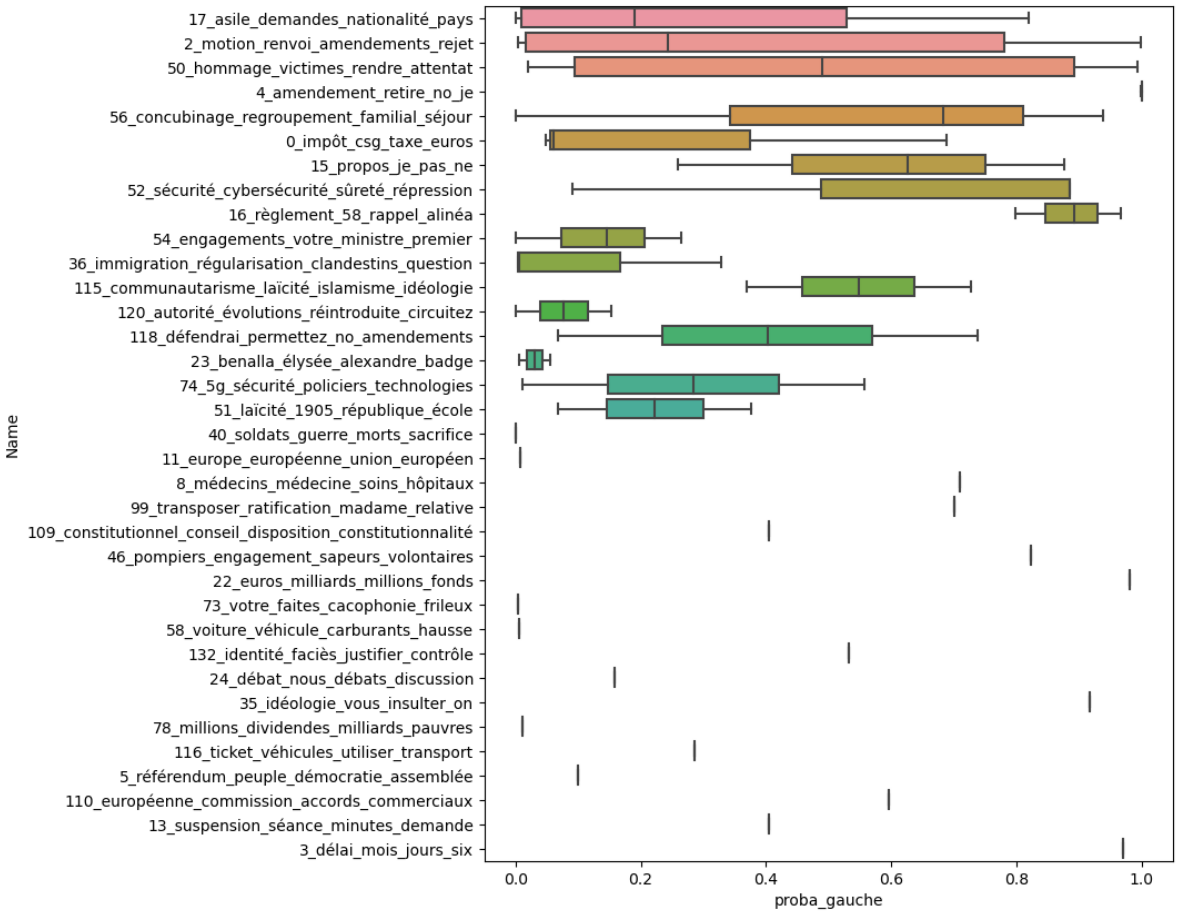
Appendix 13 – Box plot for the deputy Marine Le Pen : Position on her main topics



Appendix 14 – Box plot for the deputy Adrien Quatennens : Position on his main topics



Appendix 15 – Box plot for the deputy Jean Lassalle : Position on his main topics



Appendix 16 – Box plot for the deputy Eric Ciotti : Position on his main topics