Classification of political line
Machine Learning for Natural Language Processing 2020

Aymeric Floyrac
ENSAE
aymeric.floyrac@ensae.fr

Benoit Ané
ENSAE
benoit_ane@ensae.fr

Abstract

Our objective is to classify political lines (far-left,left, center, right or far-right) of short sentences based on a selection of political tweets from the French 2017 presidential election. We obtained good results with our benchmark models and tried to outperform these models with a Deep Learning approach. We added an interpretability part where we explore the attention mechanism of a CamemBERT-based neural network. See our GitHub or our Colab.

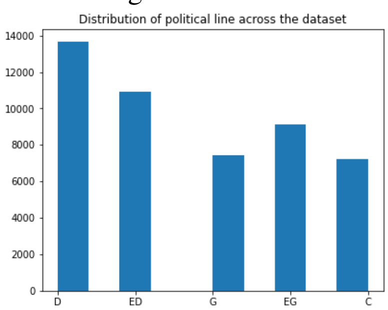
1 Problem Framing

Our main objective was to predict the political line of a short text like a political tweet. In order to do it, we used political tweets from French 2017 elections (from (Gaumont et al., 2018)). Once the data cleanup completed, we built several simple models as benchmark before training more complex models.

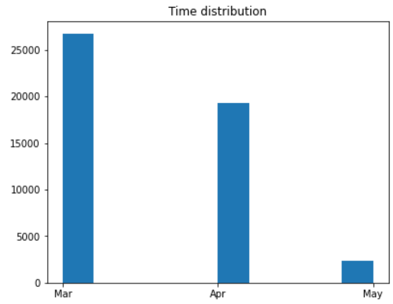
2 Experiments Protocol

2.1 Data

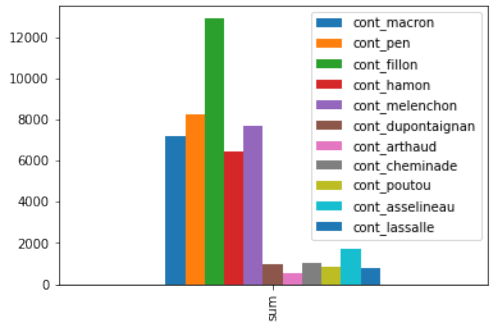
We use tweets posted during the French 2017 presidential election. These tweets come from 1) official candidates' accounts or 2) others accounts but declared as a retweet from official candidates' accounts. Finally, we have 48 377 tweets divided into 5 categories as showned in following graph.



Each category is well represented. We can also show the monthly distribution. It is very important to consider the date in our predictive models due to hot topics during the election campaign. Most of our tweets have been posted in March.



In addition, we can also show the distribution over the different candidates. We see that our dataset seems to be representative of the reality.



Finally, we can present the time distribution of political tweets considered in our dataset.

	EG	G	C	D	ED
Mar.	5 269	4 834	2 752	8 382	5 504
Apr.	3 609	2 402	3 604	5 229	4 426
May	243	200	861	57	1 005

2.2 Models

we used several types of models. As benchmark models, we considered a simple model: a multinomial logit (ML), and more sophisticated models:a multilayer perceptron (MLP), a support vector machine (SVM) and a random forest (RF). Next, we built an Recurrent neural network (LSTM) and also a Convolutional neural network (CNN) ((Wang et al., 2018) and (Yin et al., 2017)) to compare the performances of deep learning models. Finally, we fine-tuned a CamemBERT-based (Martin et al., 2019) classifier (C) in order to build our strongest model. Our catalogue of models allows us to explore several methods and to compare their performances. The Deep Learning approach has been inspired by a large literature (Iyyer et al., 2014).

2.3 Implementation

We implemented the benchmark models with Sklearn. All Deep learning models are coded in PyTorch, and sometimes enhanced with learning rate schedulers or early stopping procedures.

2.4 Model training

We splitted the data set into three parts: train, validation and test. As pointed out in part 2.1, political discourse is highly contextual and depends on hot topics. Therefore, we performed two types of split: a random split, and a time-wise split. For the latter, we use March and early April as train set and the rest of April and May for the validation set, and the remainder as test set. The random split is the classical approach, however it may lead to an overestimation of our models performance by showing them all the topics. This situation is unrealistic, and what we would want in reality is to build a classifier that would be robust to new topics and contexts. The time-wise split also shows how important the context is to classify political line. Concerning the LSTM and CNN models, we tried several architectures with different configurations of parameters. For CamemBERT, we tried to fine-tune the general model to our specific classification problem.

3 Results

We evaluate our models on an test set. In order to evaluate our models, we focus on 4 metrics : accuracy (A), recall (R), precision (P) and F1-score (F1). The following table presents the performances of benchmark models (recall, precision and f1-score are taken as averages over classes).

Split	Random				Time			
Model	A	R	P	F1	A	R	P	F1
ML	0.97	0.97	0.97	0.97	0.63	0.74	0.62	0.58
SVM	0.97	0.97	0.97	0.97	0.03	0.2	0.01	0.01
MLP	0.97	0.97	0.97	0.97	0.63	0.73	0.60	0.58
RF	0.94	0.93	0.95	0.94	0.28	0.48	0.68	0.38

We can see that the performances of the models are quite correct. Concerning our RNN and CNN models, the next table sums up its performances.

Split	Random				Time			
Model	A	R	P	F1	A	R	P	F1
RNN	0.91	0.90	0.91	0.90	0.31	0.44	0.38	0.30
CNN	0.92	0.92	0.92	0.92	0.42	0.49	0.43	0.39

We see that the neural networks have similar performances and do not outperform the benchmark models. Focusing on our CamemBERT model, we present its performances in the following table.

Split	Random				Time			
Model	A	R	P	F1	A	R	P	F1
C	0.96	0.96	0.96	0.96	0.68	0.73	0.59	0.61

4 Discussion and conclusion

We can see that outperforming the benchmark models is not easy. Simple models seem to be a good solution in order to predict political line. Moreover, we showed that designing a robust model is a complex task. Also, we showed that the results are heavily dependant on the selection of the training and test sets, which indicates that there are few cross-context indicators of political line in a sentence. Concerning the data, we found some tweets like ”il faut se rassembler autour de notre candidat” which is a sentence that every political party can tweet. This decreases our models performances.

5 Conclusion

We managed to build a political line classifier with acceptable performance, however we have shown that it may have a hard time facing a new political context. The question whether it is possible to build a dataset large enough to take into account any political context is to be studied.

References

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122.

Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. 2017. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*.

Shiyao Wang, Minlie Huang, and Zhidong Deng. 2018. Densely connected cnn with multi-scale feature attention for text classification. In *IJCAI*, pages 4468–4474.

Noe Gaumont, Maziyar Panahi, and David Chavalarias. 2018. Reconstruction of the socio-semantic dynamics of political activist twitter networks—method and application to the 2017 french presidential election. *PloS one*, 13(9).

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *arXiv preprint arXiv:1911.03894*.