Who wrote this : a framework for French novelist identification
Machine Learning for Natural Language Processing 2020

Romain Avouac
ENSAE
romain.avouac@ensae.fr

Margot Eteve
ENSAE
margot.eteve@ensae.fr

1 Problem Framing

Being able to identify authorship of a document is beneficial for a wide range of applications. It can prove very valuable from an historical perspective, since archives worldwide are full of documents whose authorship is not known with certainty. Furthermore, multiple plagiarism cases in literature could be solved with such an algorithm[1]. Against that background, this project aims at developing a natural language processing (NLP) pipeline for authorship identification.

2 Experiments Protocol

2.1 Data.

We choose to limit our experiment to a selection of ten French novelists from the 19th century[2]. Several reasons motivate this choice. First, the 19th century features coherent and identifiable literary movements, whereas the 20th century literary is much more scattered for instance. Besides, the language used in 19th century books appears close enough to contemporary French, enabling the use of embeddings pre-trained on modern corpora. Finally, 19th century books are now in the public domain, and importantly directly available in digital format thanks to the *Project Gutenberg*[3]. For each author, three representative books have been selected : two for the training set and one for the test set. This clear separation between train and test sets ensures that we are not using a given book peculiarities (e.g character or place names) for prediction. Each book has been cut into paragraphs and each paragraph has been associated its author as label. This unit of analysis has been chosen because it is large enough that it can contain substantial statistical information and yet small enough that it can be processed easily by most NLP algorithms. The construction of the train and test sets can be fully reproduced using the scripts available in the GitHub repository of the project[4].

2.2 Preprocessing.

A major advantage of using data from the *Project Gutenberg* is that it is highly normalized and virtually noiseless. It thus requires very little preprocessing to be directly usable in a machine learning pipeline. First, we perform a tokenization step. However, as we seek to distinguish authors based on their writing style, results can be highly sensitive to the choices made at this step. For instance, the amount of punctuation and the way it is used can be very distinctive of an author's style, yet it is generally removed by standard tokenizers. In order to ensure robustness of the results, we use various tokenization heuristics.

2.3 Models.

We compare the performance of several models on the classification task. Our baseline consists in a linear classifier (logistic regression) trained on the TF-IDF weights matrix. All the other approaches we leverage are based on continuous word representations. The main challenge in this case is the computation of document vectors – paragraph vectors in our case. Since there aren't actual theoretical guidelines to do so, we compare various models. First, we use a fully unsupervised technique : paragraph vectors are computed as a simple average of `FastText` [1] pre-trained French word vectors. Then we implement `doc2vec` [2], a supervised approach designed to learn word and document vectors jointly. For these two approaches, classification is also performed by training a linear

[1] See here and here for historical examples.

[2] Zola, Maupassant, Daudet, Stendhal, Balzac, Flaubert, Hugo, Dumas, Vigny and Verne.

[3] https://www.gutenberg.org/

[4] https://github.com/meteve/NLP_project

classifier on the paragraph vectors matrix. Finally, we use the `CamemBERT` model [3], a `BERT` model pre-trained on a large French corpus. The model is fine-tuned in a supervised way through a simple dense layer with softmax activation.

2.4 Evaluation.

We choose to evaluate our models with the F1 metric, which puts equal emphasis on both precision and recall. We also provide a more qualitative analysis of performance by analyzing the confusion matrix.

Both the NLP and the classification models are fine-tuned using grid search on a validation set which consists in half the test set. The other half is used for final evaluation.

3 Results

Performance comparison of the different models is presented in table 1. All the steps needed to reproduce these results are presented in a Colab notebook[5].

First, we observe that the TF-IDF approach is, as often, a strong baseline (F1 = 0.44). Since we deal with 10 classes, a random classifier would generate a F1 score of 0.1. Thus, TF-IDF results appear quite satisfactory for such a simple approach. However, we notice that it tends to greatly overfit the training set (F1 = 0.9), likely because of the presence of peculiar character and place names which makes separation too easy on the training set. To support this intuition, we analyze the TF-IDF weights for the well-classified paragraphs of the training set for each author, and indeed find some character and place names in the top weights. We use Named Entity Recognition to detect and remove these elements. The F1 score on the training set is lower after removing the named entities (F1 = 0.82), but still high. Surprisingly, we find that the score on the validation set decreases (F1 = 0.42). The named entities seem to be relevant to identify authorship, since some characters and places can appear in multiple books. We thus decide to keep the proper nouns in the models.

Second, we observe important heterogeneity between the approaches based on continuous word representations. The FastText model performs the worst (F1 = 0.35), likely because it fails to take into account any context information due to the naive averaging procedure. The Doc2Vec model produces better results (F1 = 0.39), but still performs relatively poorly. On the contrary, the CamemBERT model exhibits much better performance (F1 = 0.5) and turns out to be the only approach which beats the TF-IDF baseline.

Table 1: Comparison of F1 scores on the test set

Model	**F1 score**
CamemBERT	0.5
TF-IDF	0.44
Doc2Vec	0.39
FastText	0.35

Finally, we notice a strong heterogeneity between authors through the analysis of the confusion matrix, displayed in figure 1. CamemBERT well classified 15.6% of Zola's paragraphs, whereas it well classified 87.2% of Dumas's paragraphs. The results between the two authors are reversed when it comes to precision, and the gap is reduced: 91.8% for Zola, and 78.7% for Dumas. Dumas or Flaubert seem to be more easily identified than Daudet or Vigny. Some authors may have a more singular writing style than others, or simply reuse character and place names in several books.

4 Discussion

Our results show that it is possible, to a certain extent, to perform authorship identification of French 19th century novelists using NLP algorithms. However, we find that the performances of complicated deep learning approaches only come close to those of a simple TF-IDF model with linear classification, which is disappointing. However, our study has important limitations. In particular, due to the high computational cost of fine-tuning the CamemBERT model, we had to set a maximum number of tokens per paragraph of 100, and we couldn't properly tune the learning rate. Without these limitations, it is likely that this approach would have produced more satisfactory results.

An idea for future works could be to use the tools of *stylometry*[6] – which is used to attribute authorship in practice – in order to build relevant features based on field knowledge. Such an approach could help to better distinguish authors based on their writing style.

[5] `https://colab.research.google.com/drive/18Hj2zI-dL2YjCyW4vqPSkLYzNLRmk3wD`

[6] see here for more details

References

[1] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[2] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[3] 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. *arXiv preprint arXiv:1911.03894*, 2019.

Appendix

Figure 1: Confusion matrix

	Balzac	Daudet	Dumas	Flaubert	Hugo	Maupassant	Stendhal	Verne	Vigny	Zola	RECALL
Balzac	74.000	20.000	52.000	17.000	30.000	5.000	352.000	489.00	9.000	35.000	0.068
Daudet	5.000	97.000	52.000	64.000	22.000	45.000	9.000	22.00	2.000	24.000	0.284
Dumas	34.000	6.000	2091.000	19.000	49.000	53.000	33.000	21.00	4.000	10.000	0.901
Flaubert	1.000	4.000	20.000	464.000	14.000	48.000	11.000	7.00	1.000	38.000	0.763
Hugo	10.000	31.000	79.000	48.000	202.000	23.000	1072.000	9.00	62.000	5.000	0.131
Maupassant	16.000	42.000	34.000	20.000	22.000	138.000	5.000	22.00	1.000	18.000	0.434
Stendhal	1.000	18.000	62.000	6.000	13.000	6.000	530.000	2.00	1.000	16.000	0.809
Verne	2.000	4.000	148.000	12.000	29.000	6.000	2.000	197.00	1.000	10.000	0.479
Vigny	16.000	95.000	56.000	6.000	27.000	13.000	23.000	15.00	16.000	0.000	0.060
Zola	61.000	24.000	145.000	79.000	8.000	89.000	49.000	38.00	0.000	485.000	0.496
PRECISION	0.336	0.284	0.763	0.631	0.486	0.324	0.254	0.24	0.165	0.757	NaN