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 ; for instance, the authorship of some of Moliere or Shakespeare works has been debated since the 19th century[1]. Against that background, this project aims at developing a natural language processing (NLP) pipeline for authorship identification. Specifically, we focus on the task of identifying authorship of short texts written by French novelists based on their literary style.

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, in order to develop an algorithm that actually distinguishes authors based on their writing style, we need to select authors from a similar time period. 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 replicate our analysis using various tokenization heuristics. We also experiment with stemming and lemmatization techniques to determine whether normalizing data can improve performance.

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

---

[1] See here and here for more details.
[2] Zola, Maupassant, Daudet, Stendhal, Balzac, Flaubert, Hugo, Dumas, Vigny and Verne.
[3] https://www.gutenberg.org/
[4] A_REMPLIR

the computation of document vectors – paragraph vectors in our case. Since there aren't actual theoretical guidelines to do so, we employ various models and heuristics. First, we use a fully unsupervised technique : paragraph vectors are computed as a simple average of `FastText` [1] pretrained 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 classifier on the paragraph vectors matrix. Finally, we leverage transformer models, which generally achieve state-of-the-art results in most NLP tasks. Specifically, 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 very strong baseline (F1 = 0.52). 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. Against that background, we use Named Entity Recognition to detect and remove these elements. RESULTATS.

Second, we observe important heterogeneity between the approaches based on continuous word representations. The FastText model performs the

Table 1: Comparison of F1 scores on the test set

Model	F1 score
TF-IDF	0.52
CamemBERT	0.52
Doc2Vec	0.39
FastText	0.35

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 = 0.52), but doesn't beat the TF-IDF baseline.

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.

Thus, future works could be devoted to properly tune the various hyperparameters of the CamemBERT model in order to fully observe the potential of this model on such a task. Another interesting approach could be to use the tools of *stylometry*[6] – which is used to attribute authorship in practice – in order to build statistically informed features that could help distinguish authors.

---

[5]A_REMPLIR

[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.