Generation and discrimination according to authors' styles
Machine Learning for Natural Language Processing 2020

Gallier Hugues
Ensae Paris
hugues.gallier@ensae.fr

Legris Tristan
Ensae Paris
tristan.legris@ensae.fr

Abstract

In this project we focus on author discrimination and author like text generation. We thus handle two different tasks : classification of texts (the goal being to differentiate Victor Hugo's texts from other authors) ; and generation with Victor Hugo's style. As we shall see, CamemBERT seems to give very good results concerning the first task, even if a simple linear SVM is also a strong baseline. For the second task, we tried several methods, and found that a GPT2 model trained on a French corpus worked better than any other model we tried. We nonetheless also use CamemBERT to refine this generation approach.

1 Problem Framing

We think that the task at hand is particularly interesting in the sense that it requires advanced insights on the interpretation of the models to understand how it works. We would like ideally to capture stylistic characteristics of the texts.

We have collected several works of different known authors of the french literature. These authors are mainly of the 19^{th} century as we think that it is important to have comparable books in terms of language and literary genre. Indeed we have focused on novels only, rather than other genres like essays or poetry, because we are more used to reading novels. Moreover, it is less specific than the latters, so more interesting for our study. We will not provide an advanced literary explanation for our results but one may think that studying famous authors guarantees that those ones have a proper style.

For the generation, we would like our models to generate meaningful sentences (qualitative evaluation) that will be recoginzed as in the style of Victor Hugo by the models developped in the first part of our study (quantitative evaluation by a model trained to discriminate between Victor Hugo and other authors).

2 Experiments Protocol

First, a word on the data. The books were collected on internet [1] and pre-processed to turn them into proper text sequences.

As we had two distinct parts in our projet, we treat them appart in this section:

2.1 Discrimination part

In the first part we treat the discrimination problem. We would like to answer the following questions:

- Can ML discriminate authors texts
- On what features the model bases its discrimination ?
- Can we give a stylistic interpretation to those features?

We would like our classifier not to be too specific, and especially to have the following properties:

- To be robust to contextual elements (like the ordering of words)
- To be robust to the length of the text. We expect longer texts to be classified better
- To be robust to texts taken out of the scope of the training, like random text.

To ensure the first demand we will test the models not to be fooled by shuffled texts. For the second one we will train and test the models on different lengths of text. We took 3 length of text of 30, 100 and 300 words long texts. Ultimately we test the models on random text and try to make them robust to it.

[1] https://actualitte.com/thematique/16/ebooks-gratuits

2.2 Generation part

The data we collected (20 Mo of texts, which is not much but enough to have a good idea of the performances of our models) have enable us to build simple models from scratch and to fine-tune sophisticated pre-trained models like transformers.

In fact, we began by training a Recurrent Neural Network (RNN) from scratch using first vord2vec to train an embedding matrix and stacked GRU layers on top of it. We found this method to perform poorly and so decided to study other ways of performing the generation. The second approach we tried has been to use CamemBERT (Martin et al., 2020) as an embedding layer, and stacked an RNN composed of GRU layers on top of it. Similarly, the results were not satisfying, and the training took too much time in comparison of the obtained results.

We then tried to use only a fine-tuned version of CamemBERT to predict the next word to come by adding iteratively some masked tokens at the end of the text. This happened to be also underperforming. Nonetheless, we found that iteratively changing words inside a text with CamemBERT allowed for an increase in the quality of the produced samples.

We finaly used a GPT2 model trained on a French corpus (Louis, 2020) and fine-tuned it on our own corpus. We then used this model with Beam Search, and Topk/Topp sampling, and selected the best method. As said above, we used CamemBERT to refine the generation done by GPT2, and found that when used in a smart way, it could reduce the repetition that could sometimes happen in the generation of long sequences.

3 Results

Concerning the discrimination, we show thanks to an ACP that we can find an interpretation to the first components of it. However, those primary components are not sufficient to discriminate the authors through an SVM. We see that treating the text samples as bag of words and training an SVM on it achieves good results. The 2 firsts proprieties are respected, but it has some flaws. In fact, this model cannot discriminate shuffled text from real text. So this models is interesting in the sense that it is quite basic and is still pretty good. In terms or interpretation, one cannot reduce it to some components of the ACP or to the influence of some key words.

To go further, we train a deep learning model based on CamemBERT with a linear layer on it. We have to add negative samples of random text to respect the third requirement given above (in part 2.1). Once done, the model has performances a bit higher than the SVM excludes systematically random texts, while still having strong performances in discriminating Victor Hugo's texts from other texts.

As concerns the generation, we obtain some texts that are quite similar to what we could find in a text, but that are not perfect. In the quantitative evaluation, we were disappointed by the obtained results, but this also goes in favor of the discrimination part of our project, which was good enough to discriminate between "real" Victor Hugo and "fake" Victor Hugo.

4 Discussion/Conclusion

To conclude our project, we would like to say that with more time, we would have liked to try to use Generative Adversarial Networks. In fact, the approach we used here leads quite naturally to the parallel training of a discriminator and a generator, one trying to fool the other, that tries not to be fooled.

We found this project really interesting, and learned a lot doing it.

References

Antoine Louis. 2020. BelGPT-2: a GPT-2 model pretrained on French corpora. *https://github.com/antoiloui/belgpt2*.

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