Rap Lyrics Generation Model

”Je ne fais que raconter de la merde si vous le voulez bien entendu”

Machine Learning for Natural Language Processing 2020

Yannis Bouachera
ENSAE Paris
yannis.bouachera@ensae.fr

Mathieu Naour
ENSAE Paris
mathieu.naour@ensae.fr

Abstract

This project aims to build a pipeline to train a Natural Language Processing model to mimick the songwriting of one or several artists. After data scraping and some exploratory analysis of the obtained song lyrics, two models are deployed and presented : one simple but underwhelming LSTM network as well as a fine-tuning of *french GPT-2*. Two types of generation are tried : from scratch and from a user-prompted sentence initialization. We evaluated our results qualitatively by asking people to guess whether a sentence has been written by the rapper we picked or our model.

1 Problem Framing

Text generation is an interesting theme in the machine learning field, as it contrasts with other historical utilisation where a model is used to learn how to solve a problem domain given examples. Here, the goal is actually to learn properties of the domain problem and be able to generate completely new instances of this problem.

This project tackles the problem of text generation and more precisely rap lyrics generation in french. Our goal is to be able to sufficiently replicate the songwriting of a french rapper and be able to generate a full synthetic song with the help of Natural Language Processing.

The french rapper *Booba* was picked, as it is one a the very few french rappers with decades-long career and several hundred songs to his credit, while maintaining a songwriting *relatively* coherent and meaningful throughout his songs (while other french rappers with huge discography do not have the same property).

Lyrics were scrapped on the *genius.com* website using their dedicated API. The cleaning and analysis step of the obtained lyrics is fully presented in the notebook, and quickly summed up in section 2.

We chose to study two different approaches to this problem. First, by using a relatively modest model such as a LSTM network, our goal is to deploy a solution from scratch that we could fully handle and understand, even though it would probably be insufficient to successfully solve our task. Later, by using a consequently larger model (french variation of *GPT-2*), our goal was to try and finalize its training on our corpus and generate new lyrics. We will present our results and compare them in section 3.

2 Experiments Protocol

General characteristics of the scraped lyrics was performed, to grasp the structure of our dataset. While containing a lot of vernacular words as well as some *verlan*, the general sentences resembles usual french. Data cleaning mostly was mostly focused on removing some bad ”songs”, misclassified artifacts of the genius website, as well as removing information tags as well as backing lyrics.

Basic NLP processing was handled using the *spacy* python library. Our ambition was to deploy the LSTM model from scratch, using our deep-learning knowledge and conceive the network structure as well as write the code for it. The *GPT-2* model was implemented and fine-tuned using a comprehensive web tutorial [1].

For the second model, the hyperparameters are only slightly changed and we conducted only some experiments on the temperature of the model.

[1] https://reyfarhan.com/posts/easy-gpt2-finetuning-huggingface/

3 Results

Sadly, the implementation of the simple LSTM model suffered numerous major drawbacks. We were not able to obtain satisfying results, as the output first was completely incoherent and did not contain any language consistency. After some correction of the model, we were not able to train the model at all. Constrained by time, we were simply not able to obtain a generated text for our model. Our code is still provided in the notebook.

The *GPT-2* model however provided really interesting and fun results. The good balance between overfitting and completely random generation remained hard to find, but a few sentences obtained seemed very coherent and plausible while remaining completely absent from the original lyrics, meaning the model was able to succesfulyl handle the task at hand and produce good results (among the others mediocre outputs). One example is provided as the subtitle of our report, and other good generated lines are provided here :

- *Toujours à la mode mais plus stylé.*

- *Les keufs n'étaient pas chez toi donc on peut toujours klaxonner.*

To evaluate our model, we gathered 20 punchlines generated with a model temperature of 0.8, 10 punchlines generated with a temperature of 1 (that displayed strong overfitting, but sometimes produced amazingly credible results) and 10 true punchlines from Booba. Using a form we sent to 50 ENSAE students and friends, we asked them to review each line and determine whether the line was generated or was truly written by Booba.

We obtained the results displayed in the table 1

Table 1: Percentage of good and bad classification made by our reviewers for each type of sentences.

Punchline type	Classification	%
Real lines	Correct	69.5
	Wrong	30.5
Generated ($\tau = 1$)	Correct	53.5
	Wrong	46.5
Generated ($\tau = 0.8$)	Correct	43.5
	Wrong	56.5

As reviewers disposed of vastly different prior knowledge of Booba, the results have to be interpreted with a pinch of salt

4 Discussion/Conclusion

While incomplete, our proposed solution successfully tackles the original problem by being able to handle the whole process of lyrics mimicking from scraping the data from the web all the way to outputing completely new lyric lines from zero.

The *GPT-2* model, thanks to its transformer architecture, numerous parameters and preliminary training, produced some really convincing lines. Overfitting is obviously a property one would want when trying to generate text in the style of a writer, but the good balance remains hard to pinpoint.

Obviously, the lack of output of our LSTM model is the major weakness of our work. Time being a strong constraint, we hope that this work remains enjoyable as is. Among other limits, we would like to stress the fact that the hyperparameters of our second model remained vastly untouched, and further experimentation on them could help produce better results.

As seen in (Potash et al., 2015), adding a end of verse token could also help the model understand a song structure and maybe be able to output a full coherent song instead of single lines.

Although we used *GPT-2*, we also remain lucid of the issues with such large models. Although offering amazing performance in some tasks, as advised in (Bender et al., 2021), one must remain vigilant of the dangers they hide.

References

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. GhostWriter: Using an LSTM for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924, Lisbon, Portugal. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.