Rap and Text generation
How a rap text can be generated considering the metrical, and lexical questions

E. Partouche H. Hamila N. Perdriau

Octobre 2019

Contents

1 Abstract

In this paper, we will aim to show how we can use a Long Short-Term Memory Language (LSTM) to generate a rap text or to create a “rap machine”, including the generation of diverse recurrent requirements for this kind of texts such as syllables, rhythm and a diverse vocabulary.

A large part of the previous studies on the generation of text [1] have as basis a Neural network, especially Recurrent Neural Networks (RNN): when a Neural Network is a circuit of artificial neurons made to solve artificial intelligence (AI) problems, a RNN, RNNs is a specific NNs that can handle temporal sequence and variable length input but suffers from vanishing gradient problem. It is a problem in our case because it offers a text without meaning. This is why the LSTM model appears to be more effective within the scope of text generation [2] [3] [4] : it has the possibility to correct the vanishing gradient problem of the RNN and thus learning what to remember and what to forget. An interesting work was made on the use of ghostwriting through LSTM [5] : the goal is to give the impression that a rapper has produced a new song, by reproducing his style of writing.

It is necessary to base our generation on a pre-existing set of lyrics from one or several artists on which we train our model. We have also considered the differents ways of predicting the next character, the next word or the next line. As they are several ways of doing so, we have shown some interest in the way Ruslan Nikolaev [6] [7] showed the effectiveness of the character-level model in 2018.

Still, methods to estimate the quality of generated lyrics are noted by Potash, Romanov and Rumshisky in 2015 and Malmi and his colleagues in 2016 [9], considering the style of the rapper and the rhyme density. The text has to answer to some considerations such as the occurrence of linguistic phenomena as lexis and phonetics, even if most of the time, the program elaborated appears to give a “word salad” [10] [11].

This is why using the CMU Pronouncing Dictionary appears to be essential [8], as it makes possible for us to capture the global meaning of a group of words. One of the goal of this research is to have a nicely generated pronunciation at the same time. Finally, Malmi in 2016 [9] also considers the length of verses in a rap text generated. This work, then, will be a mixture of all of these previous works, trying to produce a more precise model for rap lyrics generation, including considerations on metric and length, phonemes,

question of predictions and judgment on the uniqueness of the production. Then, we will intend to present a study of a generated ghostswritten text based on some Kayne West's lyrics, considering the accentuation on syllables of this text and the metric having yet partially been studied, as the lexical aspect, another of our goal being to produce a coherent new rap text, being as coherent as the Kayne West's ones.

2 Data and Methods

One of the main exercices we had to perfom was writing a model in Python, using the TensorFlow library and a RNN, that would take into account our needs. In it, we have added the CMU Pronouncing Dictionnary, allowing us to read the generated text produced with the good accents, considering the word's syllable(s) and the metric of the sentence, the lexical stress playing also a role in the accentuation of words. The intial set of lyrics came from Rap Genius, where we downloaded the lyrics thanks to a scraper [12], hoping to produce ghostwritten texts, as if Kayne West had produced a new song. In this way, we calculated the similarity between the verses generated and Kanye West's verses, in order to see if our generated lyrics were the"freshest" plus a calculation of syllables density and lines length enabling us to see if the produced line resemble to the artist's one, or is even better than what has produced Kanye West. Then, we trained the model on a computer with a nVidia© GTX 760 GPU card, 12Gb of RAM, and a Intel© i5-3570K CPU @ 3.40GHz for about 113.000 iterations. The training took one afternoon, while using the computer at the same time to analyse the results.

3 Results and Discussion

In the generated verses, we can observe an important quantity of nonsensical words (not existing in the training set), as in the iteration no 1000 : "kidsiin", "throuictifing", "griends" for examples. Then, the CMU Pronouncing Dictionary was not really useful for their pronunciation because it did not recognize these words, being non listed. Other words also were not recognized by the Dictionary, because they were shortened, mixed (ex : "nigga", "tryna"; it is really usual in K. W. texts) or misspelled (ex : "ocident"). On the whole, the first generated lines had not a lot of sense and the rate of similarity with Kanye West's produced verses was really low, considering the loss of the generated lines. The more we trained our model, the more the produced text were meaningful. The final iteration completely illustrate this fact : ”Now throw ya hands up hustlers, busters, boosters, hoes

[Verse 4: Kanye West] Strap kicked at me Bustack Loyee oyee or suits of the leaves Yeah I used to lose helped

[Verse 3: Desiigner] I got a plan, I spit this jeople for Beans of y’all Lookin’ at my own money, while y’all niggas packs You know the shit reasong, ah school bust you can’t connect When you gon’ end up up”

We still have nonsensical words as ”loyee oyee” (interjections in K. W. texts are also common), ”reasong” or shortened words as ”niggas”, ”lookin’”, etc. But we can clearly imagine the first line been rapped. Also, we can note that there is more regularity in the lines length than for the previous iterations. Considering this rate, even if the generated text in general is a bit shorter than the lines of Kanye, the length still remains close from the original lines length (the average for both of them turns around 4,1 - 4,3 words). We even have inidications on who is rapping. This is why the more we train our program, the more the rate of similarity is close from 0, meaning that between the verses’ corpus from K. West and our verses’ corpus, there are fewer and fewer losses. Because of the non-identified words problem, we may have the feeling that the lines in the generated text contain less syllables than the original one. Nevertheless, we observe that the general average of syllables in the generated lines turns around 8 syllables per line while the one of the original text turns around 9.3 syllables per line : they are still very close. Then, it could be interesting to integrate in this kind of program a dictionnary that recognizes not the word as a whole (which is important for the pronunciation of the generated text) but first every syllables of every word, even if it has not sens or countains mistakes in its spelling. It may also be interseting to create a program with rhyming constraints.

References

[1] I. Sutskever, J. Martens, and G. Hinton, “Generating Text with Recurrent Neural Networks,” p. 8.

[2] F. Chollet, “Deep Learning for humans,” Dec. 2018.

[3] J. Raiman, “Nano size Theano LSTM module,” Mar. 2015.

[4] G. Surma, “Text Predictor - Generating Rap Lyrics,” *Toronto, Ontario*, Oct. 2018.

[5] P. Potash, A. Romanov, and A. Rumshisky, “GhostWriter: Using an LSTM for Automatic Rap Lyric Generation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 1919–1924, Association for Computational Linguistics, 2015.

[6] R. Nikolaev, “Generating Drake Rap Lyrics using Language Models and LSTMs,” Apr. 2018.

[7] R. Nikolaev, “Drake-lyric-generator,” Apr. 2018.

[8] H. Hirjee and D. Brown, “Using Automated Rhyme Detection to Characterize Rhyming Style in Rap Music,” *Empirical Musicology Review*, vol. 5, no. 4, pp. 121–145, 2010.

[9] E. Malmi, P. Takala, H. Toivonen, T. Raiko, and A. Gionis, “DopeLearning: A Computational Approach to Rap Lyrics Generation,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, pp. 195–204, 2016. arXiv: 1505.04771.

[10] H. G. Oliveira, “Automatic generation of poetry: an overview,” p. 7.

[11] F. Paupier, “How to build and deploy a lyrics generation model — framework agnostic,” Sept. 2018.

[12] F. Paupier, “RapLyrics-Scraper,” Sept. 2018.