

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

E. Partouche & N. Perdriau

CLILLAC-ARP, 7 rue Thomas Mann, 75013 Paris, France

IRIT, 7 rue Thomas Mann, 75013 Paris, France

{nelson.perdriau,eitanite.partouche}@etu.univ-paris-diderot.fr

RÉSUMÉ

Dans cet article, nous chercherons à montrer comment nous pouvons utiliser un langage de mémoire à long terme et à court terme (LSTM) pour générer un texte de rap ou créer une "machine à rap", y compris en prenant en compte diverses exigences récurrentes pour ce type de textes comme les syllabes, le rythme et un vocabulaire diversifié.

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.

MOTS-CLÉS : Apprentissage par machine, linguistique, rap, LSTM, python

KEYWORDS: Machine Learning, linguistics, rap, LSTM, python

A large part of the previous studies on the generation of text 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, still composed of Neural Networks, can remember them because they are recurrent but encounters some limits. This is why the LSTM model appears to be more effective within the scope of text generation : 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 : the goal is to give the impression that a rapper has produced a new song, by reproducing his style of writing.

One of the main exercices we had to perform 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 Dictionary, 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. Then, we trained the model on a computer for about 113.000 iterations.

In the generated verses, we can observe a important quantity of nonsensical words, as in the iteration n°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. The more we trained our model, the more the produced text were meaningful. The final iteration completely illustrate this fact... The poster will discuss the output productions of the LSTM. The project was supervised by Nicolas Ballier and Jean-Baptiste Yunès at Université de Paris