

CS 108 Project Proposal

Rhyming Automatic like I'm in Illmatic

Kevin Akos Rankine
rankine@college.harvard.edu

November 8, 2016

1 Introduction/Motivation

Hip Hop/Rap music has become the driving force behind youth culture in the United States since the mid-1990s. Artists like Jay-Z, Kanye West, and Drake are pop culture icons while deceased artists like Tupac and Big have been immortalized in the hearts of those that love their music. At the core of hip-hop lies a focus on being able to string words together in a manner that flows in terms of rhyme structure (particularly in terms of assonance and consonance) while at the same time conveying a message. This is a difficult task, as evidenced by the profusion of rappers one might call 'whack' or 'corny'. One might very well inquire as to whether this is a task that a natural language processing system might be up to in order to help these rappers with ailing careers.

My goal with this project is to investigate and implement a basic NLP system to generate bars ('bar' refers to a single line in a rap song). This problem can be tackled at different levels of granularity and scope - conceivably one could implement a system that is able to generate full-fledged songs from scratch based on the specification of a topic. However, given that the time and resources we have for this project are limited I intend to constrain myself to implement a system that automatically generates a line given the previous line. Thus the final system could be used in a fashion where an aspiring rapper is writing a song, gets stuck and needs another line, inputs the last line they have so far, to which the system outputs a line that rhymes. Notably this system could also potentially be used on top of itself, in the sense that a user could query using line l_1 , the system generates line l_2 , the user inputs l_2 , the system outputs l_3 etc. The results of such an application of the system would probably be interesting but lacking in semantic coherence. It would also constitute a fairly boring scheme in that nobody wants to listen to the same things rhyme over and over.

2 Technical Aspects

This is an interesting problem from a technical perspective largely because of the fact that it requires a solution that resolves three different problems: generating lines that are coherent from a semantic point of view with both the input and themselves, generating lines that are coherent from a grammatical point of view, and ensuring that the generated lines have some sort of underlying

matching rhyming structure. Based off of this outline, we can propose a few different approaches to doing this.

2.1 Data

I already have a script that can scrape all the lyrics from a given artist's most popular albums on Rap Genius. After deciding on a suitable list of artists to use (Malmi et al. (2015) uses 100 artists), the data needs to be preprocessed in several ways including but not limited to removed very infrequent words, sections labeled 'interlude', 'intro', 'outro' etc. that are not expected to rhyme need to be excluded, word stemming etc. In addition, phoneme features are also going to be very important, so after preprocessing I'll use a system like eSpeak to extract the relevant phonemes. I'll also probably end up using some type of word vectors.

2.2 Line Ranking Approach (Question 1)

One approach for solving this problem is what I'll call the ranking approach, a more complex version of which is discussed in Malmi et al. (2015). In essence this approach involves taking a corpus of lines from various rap songs, and treating the problem as one of information retrieval. The problem to solve can thus be formulated as follows: Given a line p and a set of candidate lines Q , select the line $q \in Q$ that most 'appropriately' follows p . There are two design decisions that have to be made when implementing this type of system. First, how is the set of candidate lines generated (since evaluation on the entire corpus is likely to be intractable for most systems)? Second, how do we define one line being 'appropriate' following another line? Clearly, our notion of appropriateness must encapsulate semantics, grammar, and rhythm.

The approach in Malmi et al. (2015) divides this problem up into two subsystems (it doesn't have to deal with the problem of grammar because the lines come from a preexisting corpus) - one for checking whether a line rhymes well (using a measure primarily based on assonance) as well as a RankSVM based method that incorporates several different features of the input (including features from a neural model aimed at assessing semantic relevance based on Collobert et al. (2011)). This yields a relevancy score which is then used to select among lines that are assessed to rhyme well enough. The candidate set itself is generated using a relatively simple algorithm detailed in the paper.

The problem tackled in this paper is more complex than what I'm trying to do in that they're attempting to generate entire songs (and use more context than just the previous line to accomplish this). Appropriating the model used in this paper for my purposes involves simplifying some of the models, but keeping the core essence, the details of which I won't go into in this proposal.

2.3 Sequence Generation Approach (Question 2)

Another approach for solving this problem is to take the input line as context, and generate the output line as a sequence from scratch. This is harder to get right than the ranking based approach, but is more rewarding/valuable because it generates novel lines. A lot of work has been done involving performing this sort of sequence to sequence transformation using recurrent neural networks in the past few years Sutskever et al. (2014). In essence we would be trying to make

prediction of the output sequence based on the the input sequence, as encoded in both the words in the input as well as potential phoneme-based features. I need to think a bit more on how exactly I would flesh out this approach before I attempt but it would be interesting if it was possible to design a system based on such an approach that could include a parametric tradeoff between rhyme-matching and semantic-matching. I would definitely welcome any suggestions to constrain my search/thought process.

3 Ethical Aspects

From an ethical standpoint this type of a system would be controversial in terms of questions of authorship (who is responsible for AI-generated lyrics and the effect they might have on somebody), the ethicality of a rapper writing their songs in this manner (especially given how controversial ghostwriting is in hip-hop), and in terms of the effect it may have on art as a whole - whether machine generated art diminishes art in some sense. I could also examine these questions from the point of view of which components of the system contribute to its ultimate ethicality in what manner - for instance, is there a difference between the rhyme aspect and the semantic aspect of the system in terms of their bearing on the final ethicality of the system?

4 Evaluation

The means by which I'll evaluate my final system will likely involve two at least two different measures - one to measure the quality of the rhymes in generated text and one to assess the grammatical validity of the generated text. Rhyme quality can be assessed via a measure like Rhyme Density (explained in Malmi et al. (2015)). Grammatical validity can be assessed via some sort of language model trained on the corpus itself, although I'm open to suggestions on this front (as well as the rest).

5 Teamwork

Since I'm doing this by myself I intend to do all the work. Because of how heavily technical this project is (and how much work it might take to get it right) I didn't want to put that work on someone else.

References

- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *CoRR*, abs/1103.0398.
- Malmi, E., Takala, P., Toivonen, H., Raiko, T., and Gionis, A. (2015). Dopelearning: A computational approach to rap lyrics generation. *CoRR*, abs/1505.04771.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.