JEN, A LYRIC GENERATION TOOL BASED ON GPT2 – CS 396

Jasper Gilley

jaspergilley2021@u.northwestern.edu

ABSTRACT

This paper details Jen, a lyric generation tool capable of writing song lyrics in the style of contemporary popular music. I trained an instance of OpenAI's GPT-2 model on a large dataset of song lyrics from a wide variety of musical genres. This project shows that learning models are capable of at least emulating human-created art in the realm of natural language, if certainly not emulating the reasons behind creating it.

1. DISCUSSION OF PRIOR WORK

While a considerable amount of work has been done in the realm of artistic natural language generation, likely due not least to the field's intrinsic interest for many individuals, this paper was motivated by the fact that many papers in the literature rely upon genre-specific invariants to ensure that generated lyrics are sensical. To the knowledge of this paper's author, no paper has sought to apply the advantages of deep learning (no hard-coded invariants necessary, genre flexibility) to the task of lyric generation. All of the papers discussed in the "related readings" section at the conclusion of this paper exploit some of these genre-specific invariants such as rhyme structure, which makes cross-genre work difficult and time-consuming.

2. DATASET

My dataset, found on Kaggle, consists of the lyrics of 114,723 English-language songs, in addition to a nearly equal number of non-English language songs (though these were not considered in the scope of this project.) The dataset's formidable size left ample room for filtering based on genre, which would not have been possible with a smaller dataset, which would be much more likely to be genre-specific. In the process of working on the results presented in this paper, I also developed a set of scripts for scraping lyrics from the popular crowd-sourced lyric website AZLyrics.com, though this did not factor into the final dataset, as scraping a comparable number of song lyrics would be prohibitively difficult.

3. METHODOLOGY

Some work has been done in the past by those with an interest in NLP machine learning towards developing general-purpose APIs for working with language. The most useful for the purposes of this paper proved to be OpenAI's GPT-2, as it seems to have been based on more of a general-purpose core dataset than many competitors, and is English-language only, an attribute which is shared by a surprisingly small number of its competitors. GPT-2 also has somewhat of a different architecture than many comparable language models: whereas many such models have historically been based on recurrence and convolution mechanisms, GPT-2 is based on an architecture that dispenses with these mechanisms entirely and uses attention solely for learning tasks. This architecture has been shown to yield increased performance on natural language-related machine learning tasks. GPT-2 was therefore used as Jen's primary system for working with natural language and training models for the purposes of this paper.

While it initially seemed that GPT-2's general-purpose seed dataset (trained on text scraped from Reddit, a popular social networking site) would present potential difficulties in the course of the task of generating song lyrics, this did not appear to be the case. As will be discussed further in the subsequent "evaluation" section, the word frequency of Jen's generated lyrics corresponded relatively closely to that in the fine-tuning song lyrics dataset, which suggests that GPT-2 was not "injecting" its seed dataset into the lyric generation task.

Several "sizes" of the GPT-2 system with varying numbers of trainable parameters are available to the public. For the purposes of this paper, the intermediate-sized model with 355 million trainable parameters was used. While I experimented with other sizes, this seemed to offer the best compromise between accuracy and training time.

4. TRAINING

All code for this paper was executed on the free edition of Google Colab. This provided significant time gains relative to training deep learning models locally. When Colab was set to use GPUs, the training time was not prohibitively high, so paying for Colab Pro or using the Wilkinson Lab machines available to Northwestern students was not necessary.

5. RESULTS

Given the fact that GPT-2 was trained on source lyrics which varied considerably in genre, Jen generated lyrics which could vary to a similarly wide extent. Nonetheless, it became apparent quickly that some genres worked better than others, as may have been expected. Rap lyrics seemed to work particularly well, likely because a considerable portion of the dataset consisted of the genre, and also because such lyrics tend to have strong invariants that are easily discernable by the model. The final rap dataset consisted of a composite of the total corpuses of Drake, Eminem, Kanye West, Lil Wayne, Ludacris, Nicki Minaj, The Notorious B.I.G. ("Biggie Smalls"), and R. Kelly. When trained on this dataset, the model routinely generated text such as the following (taken from a sample trained at 1500 steps):

When it rains it makin' silver raining
When it rained bullets it pimpin' tippin'
I turn this city around and I run it
I turn this city around and I run it I was sick of them popping caper
And hiding out of town
When nobody came to party
I made my town my vacation
When nobody came to party
I made my town my vacation
When nobody came to party
I made my town my vacation I was sick of them popping caper

One can observe several invariants specific to rap lyrics appearing in such a sample. Perhaps the most obvious is the repetition of lines verbatim – this is a common theme in most genres, and rap is no exception, so we should expect to see this occur. One can also clearly observe vocabulary seemingly from the rap dataset ("rained bullets", "makin'", "pimpin'".) Perhaps most revealingly, the line endings seem to be matching up in both rhyme and theme at times, a phenomenon particular to rap lyrics ("raining…tippin'", "party…vacation.") At times, the rhyming phenomenon was more obvious, as in this sample taken from a model trained to 2000 steps:

I believe in him so baby don't tell me no lies
I believe, I believe (chorus)
I love me so, so, so, so, so
So baby don't tell me no lies (chorus)
I believe in him, I believe in me so much that I can't believe my eyes

Note the "lies…eyes" rhymes as well as the wordplay commonly found in the rap genre. The dataset included some form markings such as those denoting the chorus, which is responsible for the appearance of those markings in this sample.

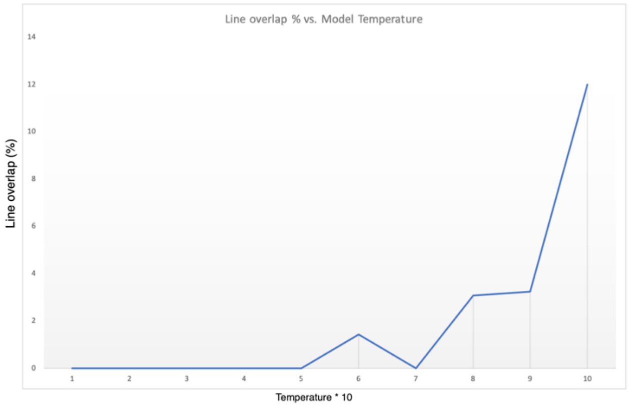
6. EVALUATION

Jen's lyric generation capabilities were evaluated quantitatively in three main ways: test-set perplexity, next word prediction, and line overlap detection.

Test-set perplexity is a measure of the predictability of wording in generated text, and is largely the *de facto* standard method for evaluating such text. Perplexity is the reciprocal of the probability assigned to the test set by the language model, normalized by the number of unique words in the test set. It is therefore essentially a measure of how much a generated corpus "fits" with the test set. Because of its reciprocal nature, lower perplexity scores are better. Jen achieved a perplexity of 278, which is slightly worse than one often finds in published papers on lyric generation (the corresponding perplexity in *A melody-conditioned lyrics language model*, one of the related readings, was 275.) However, something of an uptick in perplexity is to be expected given that the methodology of this paper relied on fewer invariants than many existing papers and was not trained solely on the fine-tuning dataset.

A closely related evaluation metric, next word prediction, was also used to evaluate the results of Jen's lyric generation system. A random line was taken from the tuning dataset and split in half, with the first half used as a prompt for GPT-2. Over the course of 50 prompts, Jen correctly predicted the next word 38.7% of the time. This figure also seems to be in line with what should be expected from a deep model trained on a dataset of this size (LSTMs trained on arbitrary data typically perform at around 40-50% accuracy on this task with sufficient training.) Indeed, that this figure is not higher than expected is an indication that overfitting to the dataset was not occurring.

To ensure further that genuine learning rather than overfitting was occurring, lines from the generated lyrics were checked to ensure that they did not occur verbatim in the dataset. At sufficiently low "temperatures" – a hyperparameter controlling how random the model's outputs are – this phenomenon did not occur at all. As temperature was increased, "plagiarized" lines grew to make up a small portion of the generated text, as can be seen here:



Percentage of lines overlapping as a function of ten times training temperature

Temperatures were subsequently kept in ranges found not to produce duplicated lines.

It should be noted that while the above evaluations were performed on the rap lyrics tuning dataset, performance was comparable on tuning datasets of comparable sizes drawn from other genres. On the slightly smaller nursery rhymes dataset, for instance, Jen achieved a next word prediction accuracy of 36.4%, with the minor drop-off likely being attributable to the nursery rhymes dataset's diminished relative size.

Generally speaking, the evaluation methods undertaken correspond closely to those pursued in published papers in the literature. In some papers – such as *DopeLearning*[2] – external human evaluation was applied to the generated text. While this is a good way of ensuring that the model is learning the correct task well, doing so quantitatively and *en masse* was beyond the capabilities of the author of this paper, for mostly budgetary reasons.

7. RATIONALE FOR PROJECT

Given the fact that this project would seek to automate a process normally reserved for creative humans, it's certainly worth considering the rationale for this project. Certainly, one can imagine that the worst human lyric-writers aren't performing any processes more complex than that which could be learned by a sophisticated learning model. Is the world really losing out if a ML model automates the lyric-writing on Justin Bieber's next hit single? The model will not be capable of writing lyrics with any degree of sophistication or genuine emotionality. "Real" poetry won't be automated away anytime soon. In a way, projects of this nature could actually increase the average artistic quality in songs, if they forced artists to differentiate themselves with lyrics that could *not* have been written by a ML model.

8. FUTURE WORK

While the findings discussed in this paper represent what this paper's author believes to be a step in the direction of relatively quick, genre-independent lyric generation, further work can of course go further towards the goal of realistic machine-generated lyrical poetry. An excellent standard for future work would be that of gaining the ability to consistently produce rhymes of the kind found in the training dataset. While papers like *DopeLearning* are able to consistently do this by hard-coding rhymes into the structure of the paper, and while the system outlined in this paper is able to do so inconsistently, a machine learning system capable of reliably exploiting this invariant in an elegant fashion would be an exciting further development.

9. CONCLUSION

This paper outlines Jen, a machine learning system capable of generating song lyrics that can demonstrate many of the invariants specific to the genre of the tuning data. Due to Jen's not being hard-coded with such invariants, she can do so in nearly any genre, provided with a large enough dataset and strong enough genre invariants onto which to grapple. With further investigation, a system capable of mimicking any genre with a high degree of reliability and accuracy seems possible.

10. RELATED READINGS

[1] Watanabe, Kento, Yuichiroh Matsubayashi, Satoru Fukayama, Masataka Goto, Kentaro Inui, and Tomoyasu Nakano. "A melody-conditioned lyrics language model." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 163-172. 2018.

[2] Malmi, Eric, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. "Dopelearning: A computational approach to rap lyrics generation." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 195-204. 2016.

[3] Bhaukaurally, Bibi Feenaz, Mohammad Haydar Ally Didorally, and Sameerchand Pudaruth. "A semi-automated lyrics generation tool for mauritian sega." *IAES International Journal of Artificial Intelligence* 1, no. 4 (2012): 201-213.

[4] Dias, Dulan S., and T. G. I. Fernando. "Komposer–Automated Musical Note Generation based on Lyrics with Recurrent Neural Networks." In *2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS)*, pp. 76-82. IEEE, 2019.

[5] Al Marouf, Ahmed, Rafayet Hossain, Md Rahmatul Kabir Rasel Sarker, Bishwajeet Pandey, and Shah Md Tanvir Siddiquee. "Recognizing Language and Emotional Tone from Music Lyrics using IBM Watson Tone Analyzer." In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1-6. IEEE, 2019.