NLP Project: Song lyrics generation using different music genres

Anonymous ACL submission

Abstract

This report describes the text generation of song lyrics. The goal is to develop a language model able to generate lyrics that are similar in style to that of a given music genre, but not identical to existing lyrics. The aim is also to provide an efficient baseline model for the goals and future directions in this domain. This paper analyzes two different language models and multiple text decoding strategies in order to develop the baseline model. The latter is then trained on four different corpus of lyrics of different genres: pop, metal, country and rap.

1 Introduction

In the past years, the field Natural Language Processing has known major advancements. Many NLP tasks are becoming easier to implement and execute. Text Generation is one of them. Deep learning techniques are being used for a variety of text generation tasks such as writing poetry, generating scripts for movies, and even for writing entire chapters of popular novels like Harry Potter. Text Generation can be architectured using different language models. Those learn the likelihood of occurrence of a word based on the previous sequence of words used in the text. They can be operated at character, n-gram, sentence or even paragraph level. In this report, Python and the concept of text generation will be used in order to build a performant machine learning model that can write song lyrics. The task is divided into three parts: dataset preparation, model training, and generating prediction. At each step, different optimization techniques are investigated and applied in order to make the model the most efficient. The performances of the models are then compared through experiments, mainly by evaluating the generated text's diversity and semantics. Furthermore, the results on data sets containing different song genres will be analyzed using the most efficient model.

2 Related work

Recent work has shown the effectiveness of Recurrent Neural Networks (RNNs) for text generation (Ilya Sutskever and Hinton, 2011). In their works, the authors use an RNN to create a language model at the character level. On top of learning a rich English vocabulary, those models learn wider dependencies such as opening and closing brackets. Those results have inspired a great amount of NLP researchers. Further studies have attempted to reproduce characteristics specific to song lyrics, such as syntax, rhythm, rhyme, and event the relation between melody and text (Ananth Ramakrishnan A, 2010; Kento Watanabe, 2018). In regards to lyrics generation based on a specific genre, Potash P., Romanov A. and RumshiskyHowever A. present a system for rap lyric generation using an LSTM (Peter Potash, 2015). Manjavacas E., Karsdorp F. and Kestemont M. have developped a Hip-Hop lyric generation using a RNN (Enrique Manjavacas, 2019). They also explored how generating text at different scales (character-level or word-level) affects the quality of the output.

3 Language models

A language model is a probability distribution over a sequence of words/characters. Two different models are investigated in this paper: an N-gram model and a Recurrent Neural Network model.

3.1 N-gram model

An N-gram language model computes the probability of a word depending on the N-1 previous words (the history). It is thus based on the concept of n-grams, continuous sequences of N words from given texts. Given a list of N-grams, the number of occurrences of each N-gram, and thus their frequency in a given document, can be computed. This enables to compute the probability function using a Maximum Likelihood Estimator (MLE):

$$P(w_n|w_1...w_{n-1}) = \frac{count(w_1...w_n)}{count(w_1...w_{n-1})} \quad (1)$$

A notable problem with the MLE approach is data sparseness. Unseen events will indeed be given probability zero, which leads to less accurate results. There are multiple methods to cope with this problem called smoothing techniques. The one that will be applied in the experiments is Laplace-Smoothing , which consists in adding a count of 1 to unseen n-grams.

The N-gram language model is based on the Markov assumption, stating that only the closest n words are relevant. This can lead to semantic issues, as this model will not be able to learn long terms dependencies in the data.

3.2 Recurrent Neural Network

A basic neural network links together a series of neurons or nodes, each of which take some input data and transform that data with some chosen mathematical function. In a Recurrent Neural Network, the outputs of layers aren't influenced only by the weights and the output of the previous layer like in a regular neural network, but they are also influenced by the previous ”context”, which is derived from prior inputs and outputs. RNNs are thus particularly useful for text processing as they can learn long term dependencies in the data (A., 2014). They can be trained for sequence generation by processing real data sequences one step at a time and predicting what comes next. RNNs can have different architectures. Two of them will be investigated in this report, namely Long-Short-Term-Memory (LSTM) and Gated Recurrent Unit (GRU). They were both chosen since their architecture is designed to be better at storing and accessing information than standard RNNs.

4 Text generation

Once the model is built and trained, the text generation can begin. This is an iterative process where a word is selected based on the sequence so far, added to that sequence and then the whole process is repeated. The task of selecting a new word based on a sequence is called language model decoding. There are multiple decoding strategies strategies to pick the next word. Depending on the decoding strategy, the generated text diversity and grammar can vary. The different ones investigated here are greedy, sampling, top-k sampling and sampling with temperature. (Ari Holtzman, 2020).

4.1 Greedy decoding

Greedy decoding is the most common decoding strategy. This strategy is a maximization process, where the selected word is always the one with the highest likelihood. If this is the most commonly used strategy, the results show that it is not always lead to high quality text. One way to introduce a bit more diversity is by selecting randomly from the k words having the highest probability.

4.2 Sampling decoding

Sampling decoding is simply applied by selected the next word by sampling based on the conditional probability distribution.

4.3 Top-k sampling

At each time step, the top k possible next tokens are sampled from according to their relative probabilities. Given the original distribution, the k words with the highest probability are selected. The distribution is then re-scaled by the lowest probability of those k-words and the new words is selected by sampling based on this new distribution.

4.4 Nucleus sampling

Nucleus (or top-p) sampling uses the shape of the probability distribution in order to determine the set of words to be sampled from. The cumulative distribution is computed and cut off as soon as the CDF exceeds a chosen value p. This distribution is then re-scaled and the new word is selected by sampling based on it.

4.5 Sampling with Temperature

Finally, the last used approach is temperature sampling. This strategy is generally applied when using Recurrent Neural Networks. The probability distribution is scaled by a coefficient (the temperature) before applying the softmax function. A low temperature leads in high probability results, while a higher temperature leads in more surprising ones.

5 Evaluation method

Text generation is a task which can not be acutely evaluated. Due to the subjectivity and of the task, there is currently no evaluation method fully measuring the quality of a generated text. This applies all the more for song lyrics generation, as it is a very creative domain and the generated sentences do not always have grammatically correct. However, there are some metrics that can give insight in order to compare different methods. The two metrics used for evaluation in the experiments are perplexity (PPL) and Self-BLEU score. They are both used to evaluate the generated text's diversity.

5.1 Perplexity

The perplexity is a measurement of how well a language model predicts the test data (Bahl and Mercer, 1997). The perplexity (or PPL) of a language model on a test set is the inverse probability of the test set, normalized by the number of words. For a test set $W = w_1 w_2 ... w_N$, we get:

$$PPL(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 ... wi - 1}} \qquad (2)$$

Since the perplexity is computed based on the inverse of the probability, minimizing perplexity is equivalent to maximizing the test set probability according to the language model. In the case of text generation, models with low perplexity thus tend to have a low diversity and get stuck in repetition loops.

5.2 Self-BLEU score

This metric evaluates the diversity of the generated data. It uses the BLEU score, a value which assesses how similar two sentences are (Papineni, 2002). This value represents a number between 0 and 1, where 0 represents no similarity and 1 indicates that the two compared sequences are identical. For each sequence in the generated text, this BLEU score is computed for all others sequences. The Self-BLEU score of the generated text is the average of all those computed scores (Yaoming Zhu, 2018). In order to have a reference to compare the results of the experiments to, the Self-BLEU score of each song in the dataset was been computed, and all those scores were then averaged. The obtained reference score used for the experiments is 0.0322.

6 Datasets

The data sets used for the models training and testing were lyrics songs gathered from kaggle (Kaggle, b,c,a). Four different genres were compared, namely: pop, metal, rap and country songs. The datasets were extracted by grouping songs by genre. They were then were slightly preprocessed by removing all non alphabetic characters (except for spaces).

7 Experiments and results

The following section describes the experiments performed in order to get the most efficient song generator possible. It is argued that the optimal language model should produce semantically coherent text and having a Self-BLEU score close to the reference one. First, the N-gram and RNN models' architures are analyzed and optimized. Next, the different decoding strategies are evaluated using Self-BLEU and perplexity. The model giving the best results is then trained on datasets containing different song genres. Specific characteristics and vocabulary of the generated results will be analyzed.

7.1 Experimental setup

Except when specified otherwise, the models in the following section are built as follows: the RNN is character based and consits of an embedding layer of dimension 256, a GRU layer with 1024 units and a dense layer of the size of the vocabulary; the n-gram model is word-based, has size 4 and uses Laplace smoothing. The dataset used is the pop songs text file (from (Kaggle, a)).

7.2 Optimization for n-gram

This first experiment aims to find out what n-gram size produces the best results. N-grams of size ranging from 1 to 5 were trained on the 100 000 first elements of the dataset. The results are shown in Table 1. There, one can note that n-grams of bigger size result in a higher perplexity and lower Self-BLEU score. This thus indicates that bigger size models have a lower diversity (which makes sense since longer n-grams take more history into account). The model giving the closest results to a song appears to be the 4-gram, since it has the Self-BLEU score the closed to the previously computed reference (0.0322).

7.3 Optimization for RNN

As mentioned earlier, RNNs can have different architectures. Two different varieties were investigated: GRU and LSTM. After training the model on the first 1 500 000 elements of the dataset, it

Ngram	SelfBLEU	PPL
Unigram	0.470	336.598
Bigram	0.400	120.97
Trigram	0.424	182.523
4-gram	0.0279	452.252
5-gram	0.0647	617.718

Table 1: Comparison of ngram models of different sizes usinf the PPL and SelfBLEU score

appeared that the GRU architecture gave better results than the LSTM. The latter did not give semantically coherent results, even when stacking multiple layers together or reducing overfitting (by introducing a Dropout layer). For instance, the following sequences have been generated using:

An LSTM architecture:

And totheat fr milippet asiouloon mem
ser t thend in me ad
I saint men
Sing out s asperonine when l a war wit y
myoug

A GRU architecture:

Hey I don't care what they say
I don't want to live all alone
I can't conceive of the years left in me
Without you in our home

7.4 Comparing decoding strategies

In order to compare generations to the reference text, diversity was first analyzed using the Self-BLEU score. This value was calculated for each decoding method for different parameters. Results of those experiments are shown in Figure 1 and 2. One can observe that the top-k sampling and top-k greedy always produce results above the reference value, especially for smaller k's (the smaller k, the lower the diversity). This is due to the fact that those methods constantly select from most probable results. As such, similar sequences end up being predicted and it often happens that they get stuck in repetition loops. The same phenomenon happens at an even greater scale for the greedy decoding method, which has the lowest diversity of all decoding methods (as it always only selects the most probable sequence). In contrast, sampling has the lowest Self-BLEU score (0.0196), and thus, a very high diversity. Consequently, the text generated using the sampling strategy is semantically very incoherent. Reason is that even if the probability of getting words with a high probability is higher than for getting a low probability, a big part of the vocabulary is still unlikely to fit the context. This means that for each sample, the probability of getting an out-of-context word is relatively high. To cope with this problem, nucleus sampling was implemented. From the experiments using the Self-BLEU, it can be seen that a top-p sampling with p = 0.75 gives a diversity close to optimal. Nevertheless, from a semantical point of view, the generated text makes most sense at 0.95 without inducing too many repetitions. Finally, the last decoding strategy analyzed is Temperature Sampling. As previously mentioned, lower temperature result in lower diversity and vice-versa. However, while increasing the temperature increases the diversity, it comes at the cost of lowering generation quality. For the case of text generation, a temperature of 0.5 gives an optimal diversity and a sufficiently coherent text.

7.5 Models comparison

The performance of RNNs and N-grams was then compared so as to get the best song generator. In this experiment, nucleus sampling was used for n-gram and temperature sampling for RNN. From a diversity point of view, the RNN gets the Self-BLEU score the closest to the reference value but both models get satisfactorily close to it. However, semantically speaking, the RNN achieves better results. If the generated songs still contain some 'illogical' sequences, they are quite cogent for the most part. For instance, here is an example of sequences generated by the RNN model:

I want to be with you baby
And I promise not to let you go
And your heart desires will come to me

For the sake of comparison, here is an exmaple of sequences generated by the n-gram model:

I oh baby how you doing you know I'm
the type of girl I'm a world wide woman
www you can log on

Those results can be attributed to the ability of RNNs to learn longer dependencies. As the N-gram model only uses the closest N words to generate the next one, it is not able to learn those long term dependencies. It thus finds itself producing sequences which make less sense. Additionally, one can note that the RNN model even kept the song structure with the verses.

7.6 Model applied to different genres

From the previous experiments, one reasonably efficient language model can be built using the described RNN and the temperature decoding strategy with temperature = 0.5. In this section, that model will be used in order to perform songs using four different music genres: pop, rap, metal and country. For this to be achieved, the model is trained on different datasets, each containing songs belonging to a particular genre. Overall, the difference between the generated songs genre is distinctively marked. Each song genre's characteristic were found in the generated text. The metal songs generated for example often contained lyrics in capital letters (and it is commonly known that metal is generally loud and can be associated to screaming). Pop songs turned out to be most often about love stories and partying, while country songs told nostalgic stories about the sun shining on the land of America. Table 2 shows the 10 words which were the most often generated for each genre (excluding stop words, conjunctives and pronouns). This illustrates that some key characteristics of the genres are found in the vocabulary of the generated lyrics. Full lyrics can be found in Appendix A.

8 Conclusion

As a conclusion, a song generator language model was successfully built. After comparing different models and optimizations techniques, it appeared that the most efficient model for lyrics generation would be a recurrent neural network with a Gated Recurrent Unit architecture. Regarding the text decoding strategies, Nucleus sampling gave reasonably good results, but Temperature sampling is what generates the best songs, from both a diversity and semantic point of view. Training the model on different genres induced very characteristic songs, each having distinctive vocabulary. All in all, the generated songs lyrics are quite accurate, and definitely similar to actual song lyrics. Training the model longer or with more epochs could generate song lyrics with even better semantics. Further research could be done by training the models on

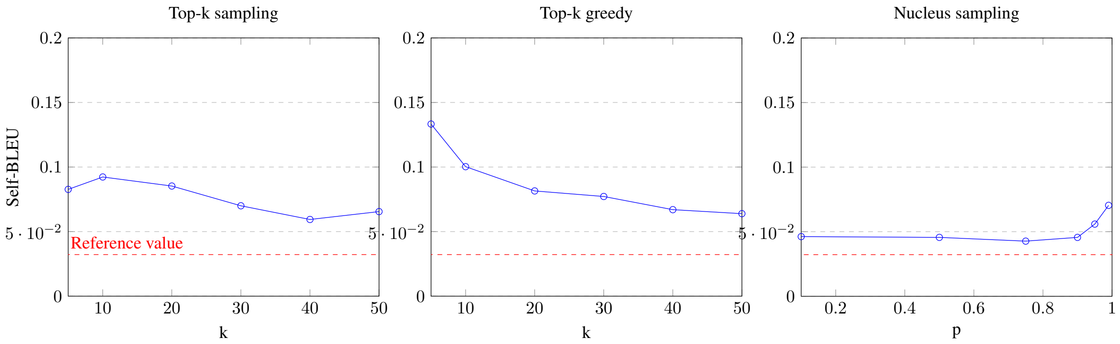

Figure 1: Self-BLEU scores of generations from various decoding methods using n-grams. The red dashed line represents the reference value computed on the dataset, i.e. the optimal Self-BLEU value.

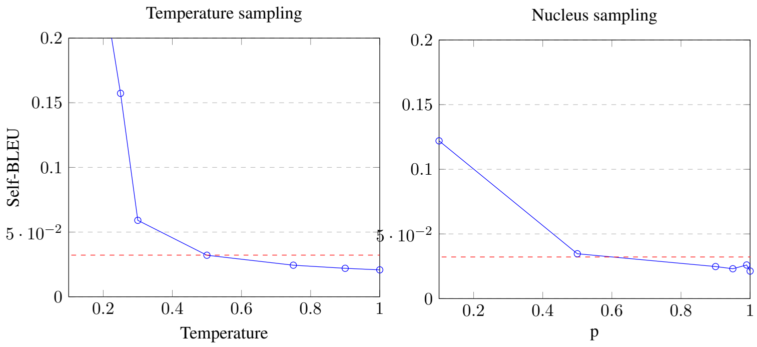

Figure 2: Self-BLEU scores of generations from various decoding methods using RNNs. The red dashed line represents the reference value computed on the dataset, i.e. the optimal Self-BLEU value.

Genre	10 most common words
Pop	’love’, ’know’, ’like’, ’oh’, ’see’, ’baby’, ’time’, ’let’, ’The’, ’feel’
Rap	’know’, ’like’, ’back’, ’never’, ’time’, ’love’, ’got’, ’say’, ’want’, ”’Cause”, , ’go’
Metal	’life’, ’time’, ’never’, ’world’, ’away’, ’like’, ’eyes’, ’feel’, ’blood’, ’pain’
Country	’love’, ’know’, ’time’, ’way’, ’heart’, ’back’, ’home’, ’away’, ’old’, ’night’

Table 2: Words generated the most for different song genres (excluding stop words, pronouns and conjunctives).

even more genre or even by linking the melody of those music genres with the text.

References

Graves A. 2014. *Generating Sequences With Recurrent Neural Networks*. Ph.D. thesis, Department of Computer Science University of Toronto.

Sobha Lalitha Devi Ananth Ramakrishnan A. 2010. An alternate approach towards meaningful lyric generation in tamil. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, pages 31–39.

Li Du Maxwell Forbes Yejin Choi Ari Holtzman, Jan Buys. 2020. The curious case of neural text degeneration. In *ICLR 2020*.

Baker J. Jelinek E Bahl, L. and R. Mercer. 1997. Perplexity: a measure of the difficulty of speech recognition tasks. In *Program, 94th Meeting of the Acoustical Society of America*.

Mike Kestemont Enrique Manjavacas, Folgert Karsdorp. 2019. Generation of hip-hop lyrics with hierarchical modeling and conditional templates. In *Proceedings of The 12th International Conference on Natural Language Generation*, page 301–310.

James Martens Ilya Sutskever and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11))*, page 1017–1024.

Kaggle. a. Drug of choice by genre using song lyrics. https://www.kaggle.com/carriel/drug-of-choice-by-genre-using-song-lyrics

Kaggle. b. Rap lyrics text mining. http:https://www.kaggle.com/rikdifos/rap-lyrics-text-mining.

Kaggle. c. Song lyrics from 6 musical genres. https://www.kaggle.com/neisse/scrapped-lyrics-from-6-genres.

Satoru Fukayama Masataka Goto Kentaro Inui Tomoyasu Nakano Kento Watanabe, Yuichiroh Matsubayashi. 2018. A melody-conditioned lyrics language model. In *Proceedings of NAACL-HLT 2018*, page 163–172.

S. Ward T. Zhu W. J. Papineni, K. Roukos. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, page 132–137.

Anna Rumshisky Peter Potash, Alexey Romanov. 2015. Ghostwriter: Using an lstm for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 1919–1924.

Lei Zheng Jiaxian Guo Weinan Zhang Jun Wang Yong Yu Yaoming Zhu, Sidi Lu. 2018. Texygen: A benchmarking platform for text generation models.

A Appendices

Here are some examples of songs generated by an RNN model with GRU architecture, trained on datasets of size 1 500 000 and using temperature sampling.

Pop song:

Oh oh oh oh oh oh oh oh oh oh
You should leave with me
I want to know that I love you like that
I would make you my will open things you know you ain’t got
more than just a little bit
If you want my body gettin’ bodied
Gettin’ bodied gettin’ bodied
You want my body
Tonight I’ll be your naughty girl I’m
Callin’ all my girls
I see you look me up and down
And I came to party
Tonight I’ll be your naughty girl
I’m callin’ all my girls
I see you look me up and down
And I came to party I’m in our home
And now the light back home
You can find it happy just to call you a can’t live

Rap song:

I was enchanted to meet you
This night is flawless

Don’t you let it go I’m wonderstruck
Dancing around all the time
I miss your things his place is too much
And that’s the way I loved you
Oohoohoohoohooh
Still love alone and never so in love
And the battle’s to skin your eyes should think about the place
Getting caught up in a moment
Lipstick on your face
And I’m so furious
At you for making me feel this way
But what can I say
You’re gorgeous
You make me so happy it turns back to sad
There’s nothing I hope out just in your face
All I can say I did words that never said
Remember the something really door song
’Cause the players gonna play play play play
And the haters gonna hate hate hate hate

Metal song:

MY BODIES DEAD AND OUT OF SIGHT
TWO WRONGS DONT MAKE A RIGHT
MY BODIES DEAD AND OUT OF SIGHT
COVERED UP WITH FRAUD
IN THE SKIES THE EVIL CRIES
YOU WILL JOIN OUR GOD
IN MY DREAMS IVE NEVER SEEN
Scream at me now I feel its so unable to fly
and through emptiness between their blood
The hand of darkness near death
These chains that burn my soul
But still you carry on
On the wings of a dream
So far from honour and pride
You say goodbye until I kill and die dry
This is my place I feel the pain inside

Country song:

I come let me know the masses
Cause I can’t let nothin’ like a fall in Tennessee
The news I could bring you good news
And I’m so bad this aint of free
Farther along we’ll know all about it
Farther along we’ll understand why
Cheer up my brother live in the sunshine
Fare you went away
And I still love you
But I don’t remember but I can’t believe it’s true
Tell me it’s not over now
Baby why you been on my mind
Because I’m free nothin’s worryin’ me
Oh I can tell by the first one to every heart that lives be over and the coffin
And he called me something told me to walk out of my life in mistouf last
Gone gone like the darkness is kind of way
Well farewell back in time theres to show to your home sweet love lifted me oh my oh
Me gotta go may a daughter still the love we’ve been the hand of the
Lord Glory glory glory