DeepLyrics: GPT2 for lyrics generation with finetuning and prompting techniques

Stanford CS224N Custom Project

Li Tian
Department of Statistics
Stanford University
lii@stanford.edu

Xiaoli Yang
Department of Statistics
Stanford University
xiaoliy2@stanford.edu

Abstract

Lyrics are incredibly important part of a song's success. The fast advancing natural language processing (NLP) methods have been successful in lyrics generation, making AI-assisting lyrics creation possible. However, as these methods get more resource-consuming and data-demanding, we identify a neglected research area of exploring efficient ways of model learning to simplify existing lyrics generation methods. Our project proposes DeepLyrics, a GPT-2 model using tuning-free prompting (in-context learning) on lyrics of highly successful songs in the past several decades to assist creative generation. Our tuning-free method, DeepLyrics, is able to achieve comparable and even better performance compared to thoroughly fine-tuned lyrics generation models. Our work shows the practicability of reducing large amount of training and finetuning work in existing lyrics generation methods with a well-engineered prompting technique.

1 Key Information to include

- Mentor:NA
- External Collaborators: Ying Lin @ Stanford Culture Lab (data preparation only)
- Sharing project: NA

2 Introduction

Good lyrics appeals to the audience emotionally, but usually it is a humanly creative process that cannot be replicated given the idiosyncrasy of cultures. With the emerging behaviors of Large Language Models, many lyrics generation methods were created to automate or serve as the first-line inspiration for creators. This hypothesis would rely on the language models' ability to learn lyrics' structure, meaning, and style to mimic the creative process.

Existing efforts are usually constrained in basic finetuning techniques with natural language prompting, so we see potential of higher-quality lyrics generation saving training/fine-tuning work with a more thoughtful finetuning and prompting design.

We will explore two main areas of training and prompting techniques: Fixed Prompt Training (finetuning and prefix-tuning) and Prompt Construction. In the process, we want to learn much about finetuning techniques for large language models, and alternative prompting regimes that would yield high quality lyrics generation. As a result of our project, we created DeepLyrics, a GPT-2 based model using prompts engineered from lyrics of highly successful songs in the past several decades for downstream generation task. It requires no fine-tuning and achieves comparable and even better performance than thoroughly fine-tuned language models.

3 Related Work

Lyrics generation, as an important aspect of people's every day literature creations, has been explored by researchers in the area of deep learning. Most research work in this area now focuses on fine-tuning the pre-trained large language models (LLMs) to their full capacity, supplementing the pre-trained model with extra information, and/or setting proper generation constraints during training for better lyrics generation. Lu et al. (2019) and Huang and You (2021) explored augmenting Seq2Seq models with a piece of original melody and melody emotions to improve lyrics generation, respectively. Ma et al. (2021) proposed AI-Lyricist, a lyrics-generating system consisting of four modules that involve a music structure analyzer, a SeqGAN-based lyrics generator trained on all parameters with multi-adversarial training, a deep coupled music-lyrics embedding model, and a polisher. As the model training and data augmentation tends more and more comprehensive, Youling (Zhang et al. (2022)), an AI-Assited lyrics creation system was launched as a functioning web app. It takes in information including but not limited to music style, rhyme, rhythm, and prompts and revisions provided by users. Although these methods present exciting and promising future steps in lyrics generation, we find a missing part in the current research to explore simple and efficient ways to generate high-quality lyrics with minimal manually supplemented information. Ventura and Toker (2022) proposed an efficient prompting technique that utilizes paraphrasing and lyrics understanding. However, it still requires training the prompting in a text summarization task which exerts extra work in preparing task ground truths (text summaries). Hence, our work fills the gap of exploring and proposing creative, efficient fine-tuning and prompting methods to simplify the existing complicated lyrics generators.

4 Approach

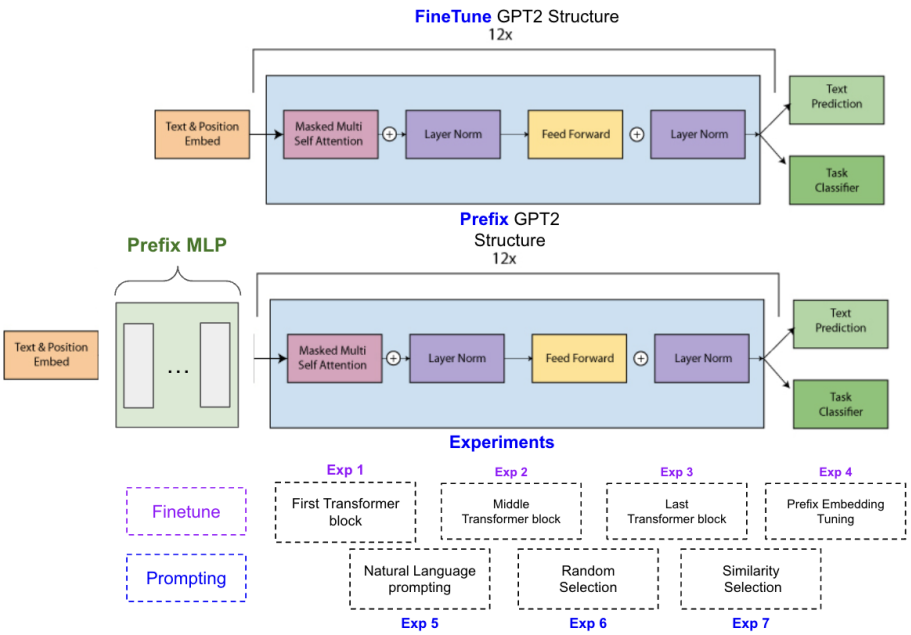

Figure 1: Finetuning and Prompting experiments on GPT2-medium lyrics generation task

We approach lyrics generation task by fine-tuning the Huggingface off-the-shelf GPT-2 model on successful lyrics. In particular, we explore optimizing the task from two dimensions - finetuning and prompting. We use the GPT-2 medium model before fine-tuning as a baseline and adopted the official Huggingface turorial and Prefix implementation to finetune GPT2 model.

Our approach will be a mix of **Promptless-Finetuning** and **Fixed-LM Prompt-tuning**, in reference to the finetuning/prompting diagram introduced in Liu et al. (2023).

- **Promptless-Finetuning**: Promptless-Finetuning refers to, with fixed prompt, finetuning the language model for downstream tasks GPT2-medium is a non-trivial model with 345 million parameters, which makes finetuning all parameters challenging. We experiment with finetuning different parts of the model to build intuition on which layer or combination of layers are more responsible for the task-specific performance.
  1. Finetune the first transformer block [1]
  2. Finetune the middle transformer block [5]
  3. Finetune the last transformer block [12]
  4. Prefix Embedding tuning Li and Liang (2021)
     Prefix-tuning prepends a prefix for GPT2 to obtain z = [PREFIX; x; y]. P_{idx} denotes the sequence of prefix indices, and we use $|P_{idx}|$ to denote the length of the prefix. In Prefix-tuning model, 2 affine layers with tanh activation is added at the beginning to embed the prefix. The sequence of modules added as as shown in Figure 2. The middle dimension is set to be mid_dim s= 512 as default.

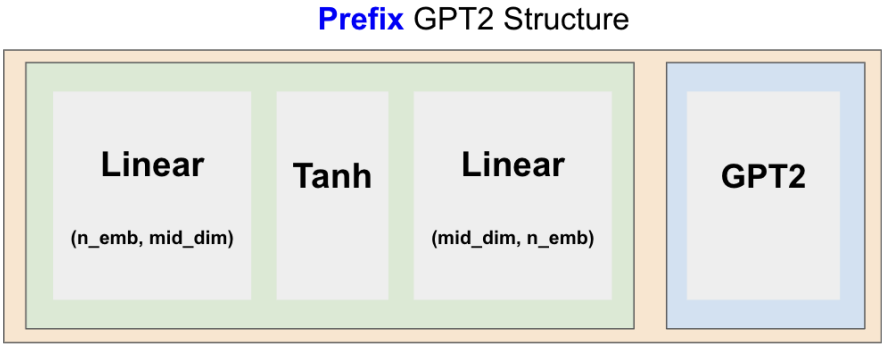

Figure 2 Prefix modification on GPT2 model structure

In training, we follow the recurrence relationship $h_i = LanguageModel_\phi(z_i, h_{<i})$, with the same training objective as finetuning $max_\phi \log p_\phi(z_i|h_{<i})$. Prefix-tuning initializes a trainable matrix P_{theta} to store the prefix parameters.Below is an annotated example of a training sample.

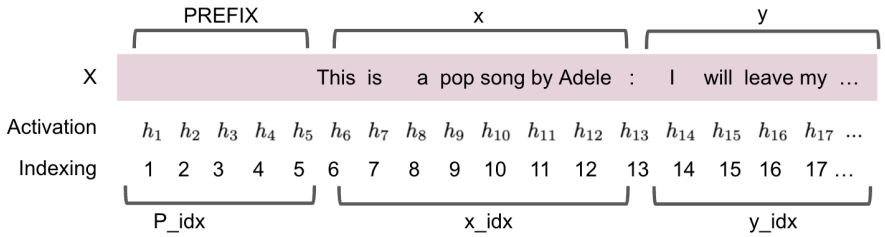

Figure 3 Prefix-tuning annotated example

- **Fixed-LM Prompt-tuning**: Fixed-LM Prompt-tuning refers to, with fixed language model parameters, searching and designing prompting structures for downstream tasks. For lyrics generation, we experiment with different prompt structures and in-context learning. Our experiment will be within the realm of "tuning-free prompting" introduced in Liu et al. (2023).
  1. Natural Language Prompt:
     i.e. *"This is a pop song lyrics by Justin Bieber:"*
  2. Random Selection: randomly select an answered example, before prompting to generate on the target
     i.e. "Rock, Breakfast Club: Every time somebody says to me ... I've gotta let my feelings show. pop, OneRepublic: Hope when you take that jump, you don't fear the fall Hope when the water rises... The only way you can know is give it all you have. *pop, Justin Bieber:"*
  3. (**DeepLyrics**) Similarity Selection: *Similarity* select an answered example from the same genre, before prompting to generate on the target:

i.e. "pop, Adele: If you're not the one for me Then how come I can bring you to your knees?... And if I'm not the one for you You've gotta stop holding me the way you do. pop, OneRepublic: Hope when you take that jump, you don't fear the fall Hope when the water rises... The only way you can know is give it all you have. *pop, Justin Bieber:"*

5 Experiments

5.1 Data

The dataset consisting of Billboard top 100 From 1958 to 2016. It records artist, genre, and full song lyrics. We preprocessed the full song lyrics by removing non-latin characters and recorded the dataset metadata below. We split the whole dataset by the Train/Val/Test ratio of 21211/4714/2357 and maintained genre and artist distribution across all three datasets. During Promptless-Finetuning experiments, we truncated lyrics at length 1024, as constrained by the max length acceptable by GPT-2 tokenizer.

Dataset Metadata

	Songs	Genre	Artist	Lyrics
Number/Avg Length (chars)	23568	5580	19	1595.9
Examples	–	R&B, pop	Justin Bieber, ColdPlay	Every time somebody says to me ...

5.2 Evaluation method

We quantify the quality of lyrics generation using three evaluation criteria: (1) Perplexity, (2) BertScores (F1, Recall, Precision), and (3) human evaluation. Human evaluation was performed by survey. Six participants were sent an artist, a genre, with a list of lyrics generated by different models, and were asked to rank the lyrics according to the lyrics' readability, coherence, and representation of the artist's style and music genre. The average rank for each model from 6 participants were then computed. A lower score represents a higher-ranked model by human evaluators.

5.3 Experimental details

For Prefix-tuning setup, we set the prefix embedding layers to be a multilayer perceptron (MLP) with middle dimension of 512 as in the original paper Li and Liang (2021), and we set the prefix sequence length as 10.

For model tuning, in both Finetuning and Prefix-tuning, we update GPT2-medium model with backpropagation using the cross entropy loss for causal language modeling against gold text lyrics, for 3 epochs ($\approx$ 7k steps) with batch size of 2s using Adam optimizer without learning rate scheduler. We save the model every 1000 steps, evaluate every 500 steps, and uses early stopping. The loss function is the cross entropy for causal language modeling, i.e. the negative log likelihood:

$$Loss = -\sum_{i}^{t} \log p_\theta(x_i|x_{<i})$$

For Fixed-LM Prompt-tuning experiments, we directly encoded different prompts using the pretrained GPT2 tokenizer and generated lyrics following the prompt texts using the pre-trained GPT2-medium model provided by HuggingFace. We truncated selected lyrics to 150 characters each to limit the total length of the prompt. We set $truncation$ True and max_length of overall prompt text to be 500 characters. Two baseline models for this section are the Prefix-tuning model and the best-performing promptless-finetuned model (Middle Block) in the previous experiments.

5.4 Results

5.4.1 Finetuning and Prefix-tuning

For both finetuning and Prefix-tuning with lyrics gold text, the training and evaluation loss consistently decreases, with no sign of overfitting. We observe that all three finetuning variations (first, middle,

last block training) reaches similar train loss. However, the evaluation loss differs significantly -finetuning the first transformer block results in comparatively worse evaluation loss, but finetuning middle versus the last transformer block of GPT2 achieves similar final evaluation loss. Comparing rate of convergence, middle-block finetuning has a slightly faster loss decrease during training than last-block finetuing.

Prefix-tuning result is slightly better than first-block finetuning. Train loss decesases at a similar rate, but the the evaluation loss plateaus compared to the finetuning counterparts.

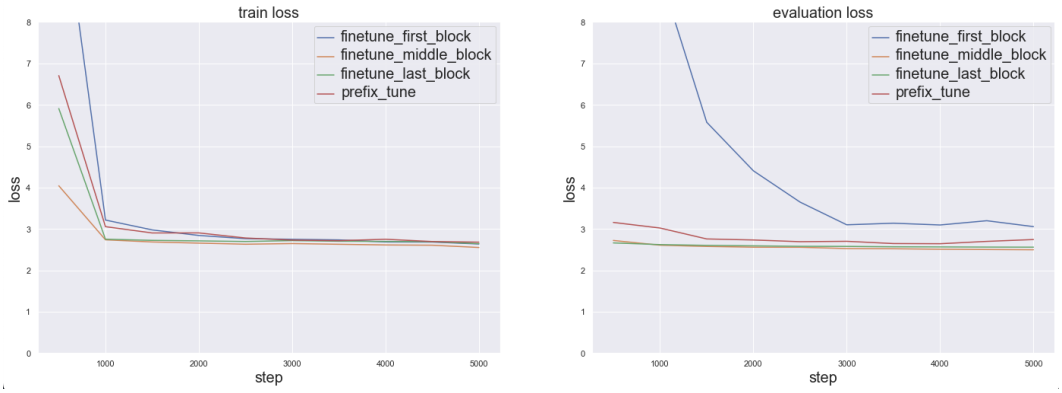

Figure 2: Train and Evaluation Loss for Fintuning and Prefix-tuning GPT2 on lyrics

Table of Experiment Results for Fine-tuning Strategies Exploration

Finetuning	Untuned (Baseline)	First Block	Middle Block	Last Block	Prefix Tuning
train loss	NA	2.6360	2.5516	2.6334	2.6801
evaluation loss	102.1126	3.0557	2.4962	2.5595	2.6421
perplexity	2.2e+44	20.6204	12.0282	12.8544	14.0426
BertScore (F1)	NA	0.781	0.785	0.784	0.780

5.4.2 Prompting and Evaluation

The three tuning-free prompting method in general achieves comparable performance across all evaluation metrics. In particular, DeepLyrics (i.e. similar selection prompt) achieves the best performance among three prompting methods as expected. It is worth noting that there is a large variance between perplexity scores achieved by different models but a small variance between BertScores and human evaluation.

Table of Experiment Results for Propmting Schemes

Test Evaluations	Middle Block (B)	Prefix Tuning (B)	Natural Lang	Random Sel	**DeepLyrics** Similar Sel
Perplexity	9.04	14.26	9.95	14.38	15.03
BertScore (F1)	0.8167	0.8119	0.7924	0.8073	0.8083
BertScore (Recall)	0.8235	0.8179	0.7992	0.8171	0.8175
BertScore (Precision)	0.8102	0.8062	0.7864	0.7978	0.7998
Human Eval (rank)	2.5	3.2	2.5	2.7	1.7

We also randomly sampled a task (Rock song, Annette) and print lyrics generation results from each model below as a reference:

Promptless-Fintuning, Middle Block
Ooh, you got that right

You better really feel it
Love the way
I touch your body

Prefix Tuning
Love iiiiiht iideiiiush baby
I want to make all her happy
That's my love Yo
wanna see me * Run from him

Fixed-ML Prompt-tuning, Natural Language Prompt
From the spirituals,
nostalgia meets Love
through some great musicianship
and not much else.

Fixed-ML Prompt-tuning, Random Sel Prompt
He was no saint
but he did help
No Reason For Hatred
had it stopped

DeepLyrics: Fixed-ML Prompt-tuning, Similar Sel Prompt
He was no saint
but he did help
us had Faith
in Jesus Christ God

6 Analysis

6.1 Fine-tuning and Prefix-tuning

The results section shows that fine-tuning the middle or the last block of transformers gives the best performance compared to the first block fine-tuning or prefix-tuning. The poor performance of tuning the first block is likely due to the long "distance" between the first block and the final generations. An adjustment made in the first block in order to improve performance will undergo all computations through later layers to exert an effect on the output, which makes the learning less effective under more variances. Fine-tuning the middle or last block is more effective for the same reason.

The prefix-tuning's underperformance is unexpected, however, as its effectiveness been proved in previous work on various tasks. One possible reason for the failure is the trade-off between model performance and number of tasks it can handle: Prefix-tuning provides the flexibility of handling multiple tasks by keeping the backbone model task-agnostic and preserving task-specific information in a set of linear layers with customizable size. It further adds flexibility with hyperparameters including $preseqlen$, $prefix\ embedding$, etc. When handling one specific task, fine-tuning directly on a particular block in the language model is more efficient as it may require prefix-tuning to have a much larger prefix network at the beginning to capture equivalent information compared to a transformer block that includes attention and historical information within itself.

6.2 Fixed-LM Prompt-tuning

According to the results section, the non-human evaluation scores between fine-tuned models (two baselines) and three tuning-free prompting models show no significant difference, but the human evaluation indicates that DeepLyrics has much more superior performance compared to others. By comparing the qualitative results from five models, we notice that the two baselines are better at capturing the genre and artist's style by using more style-specific words. Models with natural language

prompt and random selection prompt are weaker in imitating the artist's work but is more natural in its semantic meaning, which is an expected advantage of pretrained natural languages models. Finally, DeepLyrics in general presents more in-style and natural lyrics. This implies that two samples of similarly-styled lyrics are enough for GPT-2 to capture the proper style. It also has the advantage over fine-tuned baselines in its naturality. Since lyrics are usually "broken pieces of language" compared to normal sentences, this advantage likely comes from the baseline model losing the flow of natural language when it is fine-tuned on too many lyrics with un-related genre/style.

7 Conclusion

In this paper, we identified a neglected area of research in lyrics generation and explored simple but efficient ways of lyrics generation that requires less or no parameter training to achieve comparable and even better performance than training-heavy methods. We proposed, designed, and evaluated a method, DeepLyrics, that achieves superior performance in lyrics generation and requires no training or fine-tuning. We innovatively designed the Similar Selection Prompting method that takes in two pieces of lyrics in the same genre and prompts the model to generate new lyrics in that genre in style of a given artist. Our work shows the practicability of simplifying many existing lyrics generation methods to a great extent by saving training and fine-tuning work with an effective prompting (in-context learning) technique.

Inevitably, one limitation our work is that we have only compared our method to a limited number of baselines and have it evaluated by a limited number of human evaluators. Extending the method to simplify models with more complicated fine-tuning processes is an interesting future direction. For example, designing tuning-free pipelines to achieve similar performance as TRBLLmaker (Ventura and Toker (2022)). Another limitation is to evaluate our proposed method using a different backbone language model, for example, T5 (Roberts et al. (2020)).

References

Yin-Fu Huang and Kai-Cheng You. 2021. Automated generation of chinese lyrics based on melody emotions. *IEEE Access*, 9:98060–98071.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *CoRR*, abs/2101.00190.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Xu Lu, Jie Wang, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2019. A syllable-structured, contextually-based conditionally generation of chinese lyrics. In *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III 16*, pages 257–265. Springer.

Xichu Ma, Ye Wang, Min-Yen Kan, and Wee Sun Lee. 2021. Ai-lyricist: Generating music and vocabulary constrained lyrics. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 1002–1011, New York, NY, USA. Association for Computing Machinery.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Empirical Methods in Natural Language Processing (EMNLP)*.

Mor Ventura and Michael Toker. 2022. Trbllmaker–transformer reads between lyrics lines maker. *arXiv preprint arXiv:2212.04917*.

Rongsheng Zhang, Xiaoxi Mao, Le Li, Lin Jiang, Lin Chen, Zhiwei Hu, Yadong Xi, Changjie Fan, and Minlie Huang. 2022. Youling: an ai-assisted lyrics creation system.

A Appendix (optional)