Song lyrics generation

Petko Petkov, Manda Andriamaromanana, Ilyes Sais
Université Paris-Saclay, Hands-on NLP course

***Abstract*—This project explores the task of generating song lyrics given description (instruction) of the lyrics. Our approach involves augmenting a dataset of song lyrics with textual descriptions that encode mood, style, and lyrical characteristics. We fine-tune multiple Transformer-based models on this dataset and evaluate their performance through human assessments. The results show that fine-tuned models significantly outperform their base counterparts in generating coherent and stylistically appropriate lyrics.**

I. Introduction

The ability of AI models to generate creative text has seen significant advancements, with applications ranging from poetry and storytelling to songwriting. In this work, we explore the task of conditional song lyrics generation, where models generate lyrics based on structured descriptions that capture key attributes such as mood, style, and lyrical structure. To achieve this, we augment an existing dataset with textual descriptions that serve as conditioning inputs during training. We then fine-tune multiple Transformer-based language models and evaluate their performance through human assessments, comparing their ability to generate lyrics that are coherent, expressive, and stylistically appropriate.

Our primary objective is to improve lyrical coherence and thematic consistency by fine-tuning models on structured song descriptions. We focus on small Transformer models (less than 1 billion parameters) so we can quickly train and evaluate them. We compare the fine-tuned versions to the base models.

II. Dataset and preprocessing

A. Dataset overview

We use the publicly available Spotify Million Song Dataset dataset, which contains 57,650 song lyrics. This dataset does not include descriptions for the song lyrics so we introduce an additional column containing structured descriptions of the song's mood, style, and structure.

The dataset contains the following columns:

- artist - the name of the author
- song - the name of the song
- link - Spotify API link
- text - the song lyrics

Since the dataset does not contain descriptions for the song lyrics, we are generating them using large language models.

B. Descriptions generation

To generate the descriptions, we employ large language models such as GPT-4o and Gemma-2-2B. Each song is processed with a prompt instructing the model to summarize its characteristics in a comma-separated format. The generated descriptions serve as the conditioning input during fine-tuning. The prompt that we use is the following: "Describe the following song based on the lyrics in a comma-separated list of adjectives and stylistic traits (can be more complex expressions or just simple words that a person would use to describe the song). The description should include **mood, atmosphere, style, lyrical structure, and the artist's name.

Artist: {artist}
Song: {song}
Lyrics: {lyrics[:2000]}
Description:"

We started generating the descriptions with the GPT-4o model through the OpenAI API but after a few thousand samples we switched to Gemma-2-2B ([7]) because of the high costs of GPT-4o. After the descriptions are generated, we place them as a new column to the original dataset. The final dataset is uploaded on Hugging Face ([6]).

III. Model selection and training

A. Selected models

We fine-tune five Transformer-based models, each selected based on computational efficiency and performance balance. All of the selected models have less than 1 billion parameters. The models that we chose are Qwen2.5-0.5B ([9]), SmolLM2-135M (Base and Instruct), and SmolLM2-360M (Base and Instruct) [2]. The training objective follows causal language modeling, where the model predicts song lyrics based on structured descriptions.

B. Training details

Training is conducted using a batch size of 4, 8 or 16 (depending on the model size), a learning rate of 5×10^{-4} with cosine annealing, and AdamW ([4]) optimization. A 90/10 train-validation split is used, and mixed precision (fp16/bf16) is applied where possible. The fine-tuned models are then deployed on the Hugging Face Model Hub. The training time for each model is between 1 and 2 hours (depending on the model size).

The prompt that we used during the training was the following: "Generate song lyrics based on the description: {description}. Song lyrics: {lyrics}"

For the training, we are using the TRL library ([8]) from Hugging Face ([8]). It is based on PyTorch ([5]). We are logging the training progress and results to Weights & Biases ([3]) in order to gain a detailed understanding of the process.

C. Training results

The train and validation loss plots for each model is shown below (Qwen2.5-0.5B on figure 1, SmolLM2-135M on figure 2, SmolLM2-135M-Instruct on figure 3, SmolLM2-360M on figure 4, SmolLM2-360M-Instruct on figure 5).

Fig. 1: Train and loss plots for Qwen2.5-0.5B

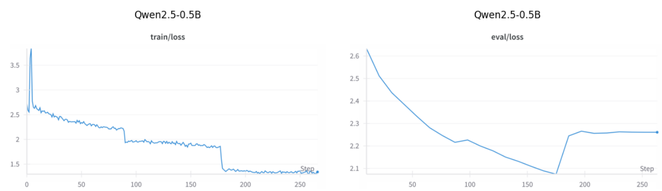


Fig. 2: Train and loss plots for SmolLM2-135M

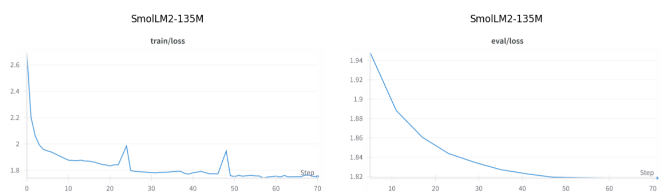


Fig. 3: Train and loss plots for SmolLM2-135M-Instruct

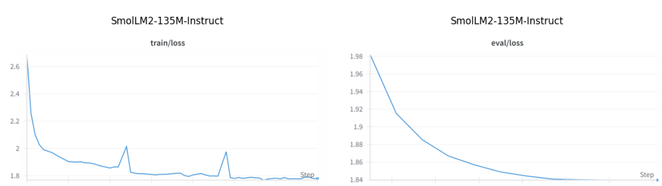


Fig. 4: Train and loss plots for SmolLM2-360M

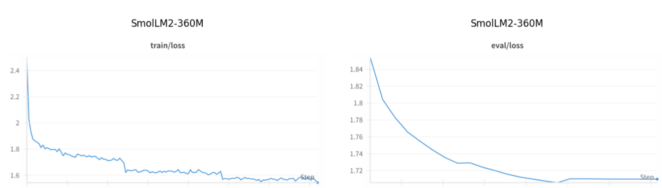
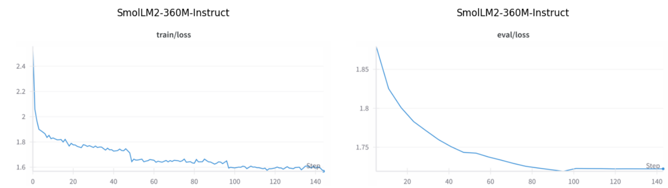


Fig. 5: Train and loss plots for SmolLM2-360M-Instruct



The fine-tuned models are uploaded to Hugging Face:

- Qwen2.5-0.5B
- SmolLM2-135M
- SmolLM2-135M-Instruct
- SmolLM2-360M
- SmolLM2-360M-Instruct

D. Compare responses on an example prompt

We compare the fine-tuned and the base Qwen2.5-0.5B models on an example prompt. The prompt is the following: ”Generate song lyrics based on the description: happy, joyful, and carefree, with a positive and uplifting vibe

Song lyrics:”

The fine-tuned model response:

I'm so glad that you're mine

I can't believe it's true

You've changed my life completely

And by telling me lies like this is no lie.

There are songs in your eyes saying ”let go” (Let go)

But they don't touch nor do for sure

So let them all fall out of reach

'Cause we'll be singing happily ever after!

Happy to say... That there ain't nothing better than knowing exactly what

you mean when you get down inside.

(Chorus:)

It makes us feel good because our hearts have never been broken before.

We will sing happily ever after forever more.

Now every day seems brighter now since yesterday night has come alive at last,

we celebrate today as if having always known tomorrow would bring something but fear into

day. So just hang up or throw away hope - cause happiness comes from finding love

everywhere around here. And though sometimes its hard to see beyond where youre bound

so lets keep dancing 'cause theres only one thing thats missing.. Is someone who may not know how

to play games instead of counting sheep well together well make lotsa money,(and then maybe wonder why.) Then Imma take some time...(just wait until Im jaded!) All right.

The base model response:

”Happy Days” by The Beatles

Verse 1: In those days when we were young, We had dreams that would never die. With love in our hearts like fire so bright.

Chorus (in English): Oh! Happy times are here again! Let's dance to this beat of ours!

(Repeat)

Bridge/Outro:

So let us sing along together now... And make every day count for joy.

IV. Evaluation and Results

A. Human evaluation

A web-based evaluation interface is developed using the Gradio library ([1]), where participants compare base and fine-tuned model outputs for 50 song lyrics descriptions. Each evaluation instance presents a song description and two lyrics generations (one from a base model and one from a fine-tuned model), allowing users to select the more coherent and stylistically aligned version. We collect results from 4 users. The interface is deployed as a Hugging Face space.

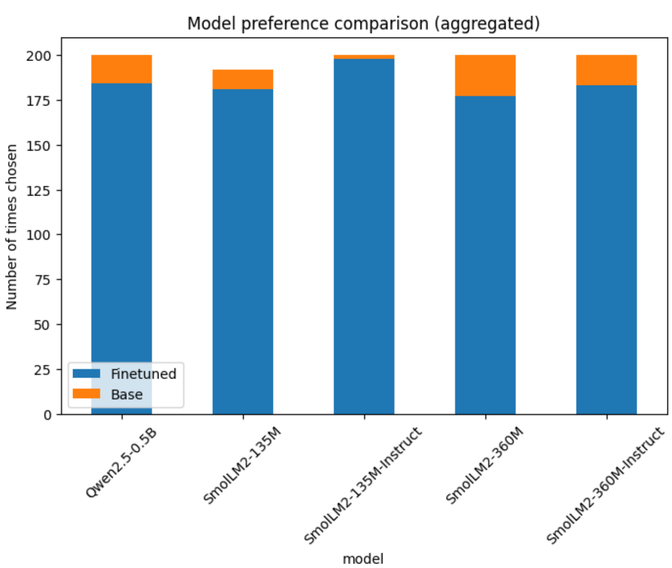
B. Results

The results indicate that fine-tuned models are preferred over base models in over 90% of cases. The SmolLM2-135M-Instruct model achieves the highest approval rating. The results are shown on table I and figure 6.

TABLE I: Model preference scores

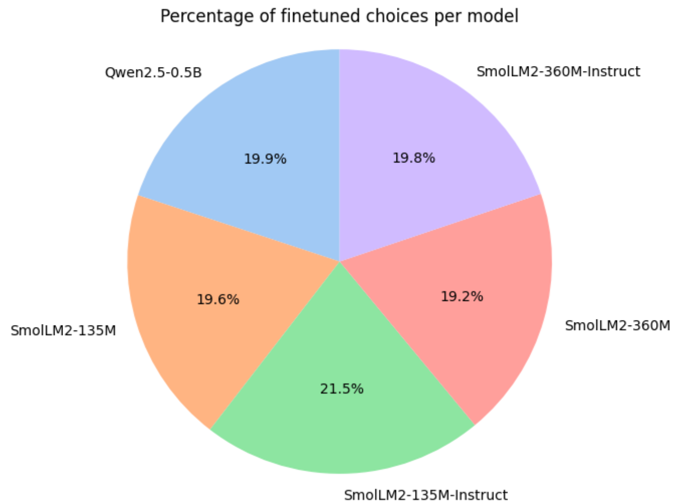
Model	Fine-tuned votes	Base votes
Qwen2.5-0.5B	184	16
SmolLM2-135M	181	19
SmolLM2-135M-Instruct	198	2
SmolLM2-360M	177	23
SmolLM2-360M-Instruct	183	17

Fig. 6: Model preference comparison (aggregated)



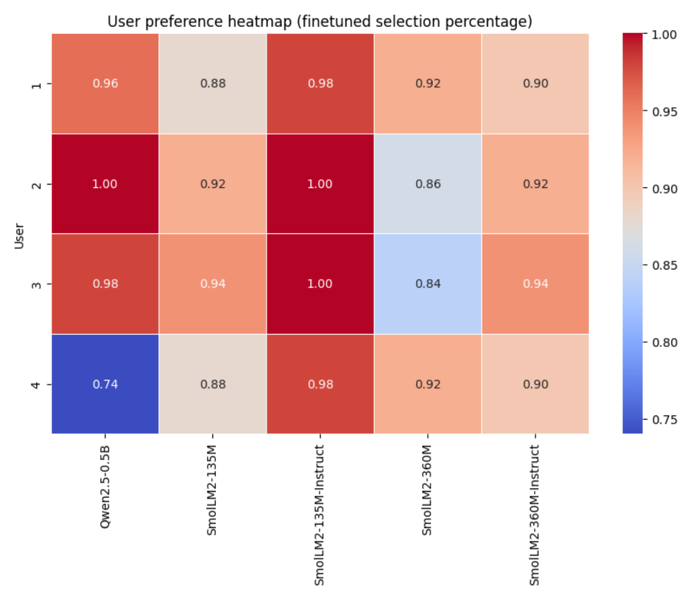
The percentage of choosing the fine-tuned model for each model pair is shown on figure 7.

Fig. 7: Percentage of choosing the fine-tuned model for each model pair



The user preference heatmap (fine-tuned selection percentage) is presented on figure 8.

Fig. 8: User preference heatmap (fine-tuned selection percenetage)



We can see that the fine-tuned versions of the models significantly outperform their base variants.

V. Conclusion and Future Work

Our study demonstrates that fine-tuning small Transformer models on structured descriptions significantly enhances the quality, coherence, and stylistic alignment of generated song lyrics. By associating lyrics with mood, style, and structure metadata, we enable models to generate more contextually appropriate and engaging lyrics compared to their base versions. Human evaluation results confirm that our fine-tuned models consistently outperform their base counterparts, with particularly strong performance from the fine-tuned SmolLM2-135M-Instruct compared to its base variant.

Despite these advancements, there are several avenues for future work. First, expanding the dataset to include a wider range of genres and multilingual lyrics could improve the versatility of generated lyrics. Second, incorporating melody-awareness—such as syllabic constraints or rhyme schemes aligned with musical structure—could lead to more musically coherent results. Third, improving the evaluation framework by integrating both automated metrics (e.g., perplexity, BLEU, ROUGE) and more extensive human assessments will provide a deeper understanding of model performance. An interesting potential evaluation is comparing the models to bigger more capable models.

By addressing these areas, we aim to push the boundaries of AI-driven songwriting, making models more adaptable, expressive, and aligned with real-world musical composition.

References

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019.

[2] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

[3] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[4] Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017.

[5] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703, 2019.

[6] Petko Petkov. spotify-million-song-dataset-descriptions (revision 763d8ac), 2025.

[7] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[8] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

[9] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.