International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Lyric Generator

Abishek Ravi [a], Ameer P Ibrahim [a], Joseph Vincent [a], Mohammed Hansif [a],Geethu Gopan [b]

[a] UG Student, Artificial Intelligence and Data Science, Viswajyothi College of Engineering and Technology, Muvattupuzha, 686661
[b] Assistant Professor, Artificial Intelligence and Data Science, Viswajyothi College of Engineering and Technology, Muvattupuzha, 686661

ABSTRACT:

In this paper, we present an innovative approach where we use the GPT-2 language model to generate song lyrics. It begins by preprocessing Eminem's lyrics dataset, adding section headers and cleaning the text. The GPT-2 model is then fine-tuned on this preprocessed dataset to learn the style characters of Eminem's lyrics. Once the model is trained, it generates new lyrics based on a given song title. This paper provides a framework for generating lyrics and offers insights into the potential of language models in music composition

Keywords: Gpt2, model,

Introduction

We have introduced a lyric generation tool which is built upon the GPT-2 language model, a state-of-the-art neural network architecture developed by OpenAI for natural language processing tasks. GPT-2 is renowned for its ability to generate relevant text based on input prompts. We start by collecting a dataset of Eminem's lyrics from various sources. This dataset is preprocessed to ensure consistency and cleanliness. The preprocessed dataset is used to fine-tune the GPT-2 model specifically for generating Eminem-style lyrics. During fine-tuning the model learns to recognize patterns in Eminem's lyrics and hence capturing his style and linguistic features. Once the model is trained, users can interact with the tool to generate original lyrics. Users provide a song title and specify generation settings such as temperature and length. The model then generates a sample of lyrics based on the input prompt. To assess the quality of the generated lyrics we conduct a similarity analysis. This involves comparing the generated lyrics to Eminem's original works by using techniques such as cosine similarity. The analysis helps users see how closely the generated lyrics resemble Eminem's style. Our tool features a user-friendly interface implemented using Streamlit for building interactive web applications. The interface allows users to input song titles, adjust generation settings, and view the generated lyrics and similarity analysis results in real-time. This model shows combining cutting-edge machine learning techniques with user interaction, our lyric generation tool offers a seamless experience for users in exploring the creative potential of AI-driven lyricism in the context of music.

Literature review

Lyrics Generation supported by Pre-trained Models

The study discusses various applications of pre-trained models, particularly focusing on the GPT-2 neural network model for the generation of song lyrics. It emphasizes the advancements in neural network architectures that have enhanced various tasks in computational linguistics, including text generation. It highlights the specific challenges associated with generating poetic texts within the musical genres like handling metaphors, metonymy, and paraphrases. By fine-tuning the GPT-2 model with English and Portuguese lyrics corpora, the study aims to analyze the quality of the generated texts in terms of spelling, syntax, and semantics. The research explores the attempt to identify patterns in the generated song lyrics and showcasing the potential of utilizing such models for creating poetic content.

Deep Lyrics: GPT2 for lyrics generation with finetuning and prompting techniques

The Deep Lyrics project introduces a novel approach to lyric generation by leveraging tuning-free prompting techniques with the GPT-2 model. The primary goal of the project is to streamline the process of creating high-quality lyrics by minimizing the need for extensive training and fine-tuning procedures typically associated with AI-assisted lyrics generation. By utilizing tuning-free prompting methods, Deep Lyrics aims to simplify the existing complex workflows of lyrics generation models. This approach involves prompting the GPT-2 model with minimal manually supplemented information, such as two pieces of lyrics in the same genre, to generate new lyrics in a similar style. The project's innovative design of the Similar Selection Prompting method allows the model to capture the proper style and produce natural and in-style lyrics without the requirement for intensive training or fine-tuning. Through experimental evaluation, Deep Lyrics demonstrates superior performance in generating lyrics compared to traditional fine-tuned models. The project highlights the practicality of reducing the training and fine-tuning workload in lyrics generation methods by effectively implementing tuning-free prompting techniques. This innovative approach not only simplifies the process of generating lyrics but also showcases the potential for achieving comparable or even better results without the resource-intensive training typically associated with AI models. The Deep Lyrics project represents a significant advancement in the field of lyric generation by introducing a more efficient and effective approach that minimizes the manual intervention and training requirements while maintaining high-quality output.

Proposed work

We use the GPT-2 language model, a state-of-the-art neural network architecture developed by OpenAI for natural language processing tasks. GPT-2 is renowned for its ability to generate relevant text based on input prompts.

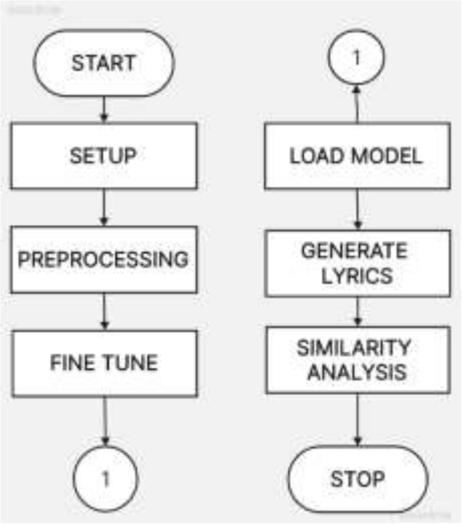


Fig. 1 – Overall process overview

Setup the working directory

Initiates the setup required for running a machine learning project in Google Colab that aims to generate Eminem's lyrics. The setup involves several key steps designed to integrate Google Colab with Google Drive, which allows for efficient management and access to project files. Specifies a root directory within Google Drive where the project's notebooks and data will reside. This is done to organize the project files in a structured manner. If this project directory doesn't exist, it is created, ensuring a dedicated workspace for the project. This setup phase is foundational, enabling seamless workflow transitions and efficient resource management for complex machine learning tasks.

Load the data

Focuses on setting up the necessary configurations and loading the dataset required for the Eminem lyrics generation project. It begins by defining the path to a json file, which is for centralizing the configuration settings of the project. By loading these settings into a dictionary, the script ensures that all parts of the project can easily access configuration parameters, enhancing maintainability and flexibility. The script retrieves the specific path for the Eminem lyrics dataset from the loaded settings. This dataset path is key to the project as it directs the script to the actual data that will be used for training the model. The data is stored in a binary format and then loaded into a Data Frame using the pickle module. This step is significant because it transitions the project from a configuration phase to a data handling phase. This transition from setting up configurations to data readiness underlines the foundational steps necessary for the smooth execution of the lyrics generation model.

Preprocess the data

Focuses on filtering and formatting the Eminem lyrics dataset to prepare it for further processing or model training. Initially, it applies a filter to select only those entries in the dataset where the lyrics contain section headers, identified by the presence of brackets. This filtering is based on the assumption that the presence of section headers likely indicates a more structured and possibly complete set of lyrics. The dataset undergoes a transformation to standardize the format of the lyrics. Each entry is prefixed with a title followed by the actual title of the song, and then the lyrics themselves. This standardization is crucial for consistency. In the final step of this process, any occurrences of triple newline characters ("\n\n\n") within the lyrics are replaced with double newline characters ("\n\n"). This step is likely aimed at normalizing the spacing between different sections or parts of the lyrics.

Download the model

The GPT-2 language model is downloaded from the Hugging Face model hub. The GPT-2 model with 124 million parameters is used. Alternatively, model with 355 million parameters can also be used, offering higher capacity and potentially more nuanced generation capabilities. This directory path ensures that the model is saved in a designated location within the project's file structure for easy access and management. The function fetches the pre-trained GPT-2 model from the Hugging Face model hub and saves it to the specified directory. This step is essential for obtaining the model parameters necessary for subsequent fine-tuning or generation tasks within the project.

Fine-tune the model

The GPT-2 language model is fine-tuned on the Eminem lyrics dataset to adapt its generative capabilities specifically to Eminem's style. The process begins by defining the model configuration parameters, including the model name, the directory path where the pre-trained model is stored, the file path to the cleaned training data, and the number of epochs for fine-tuning. The TensorFlow default graph is reset to clear any existing state and create a new computation graph. This ensures a clean slate for the fine-tuning process. Initialization is done, the TensorFlow session for training the model. During fine-tuning, the model updates its parameters based on the Eminem lyrics dataset to learn the intricate patterns and structures present in Eminem's lyrics. Checkpoints of the model's progresses are saved at regular intervals, and the training process is periodically evaluated and logged. At the end the process allows the GPT-2 model to adapt its generative capabilities to emulate Eminem's style more effectively, enabling the generation of more coherent and Eminem-like lyrics.

Load the model

The fine-tuned GPT-2 model is loaded for further use in generating Eminem-style lyrics. The process begins by defining the model configuration parameters, including the model’s name, the directory path where the pre-trained model is stored, and the directory where the training checkpoints are saved. The TensorFlow default graph is reset to clear any existing state and create a new computation graph. This ensures a clean slate for loading the model. A TensorFlow session is then started, which initializes the TensorFlow session for loading the model. During this process, the function restores the model parameters from the specified checkpoint directory, effectively bringing the model back to its state at the end of the fine-tuning process.

Lyrics generation

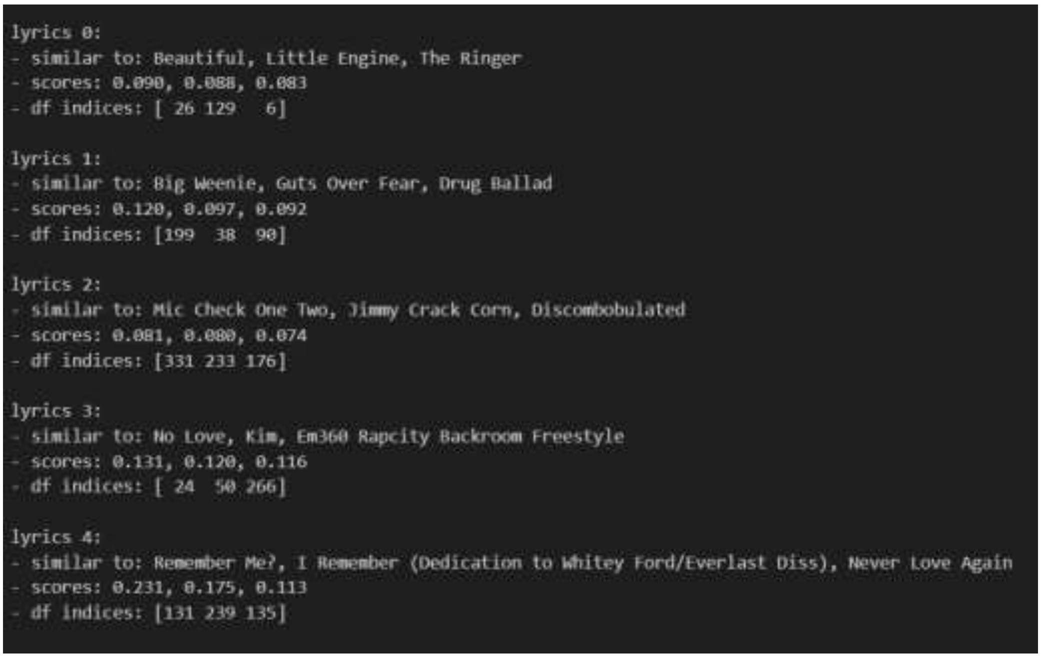
The GPT-2 model loaded into the TensorFlow session is employed to generate lyrics for a song title. Several parameters are passed to this function to control the generation process, including the checkpoint directory where the fine-tuned model checkpoints are stored, the desired length of the generated lyrics (4000 tokens), the temperature parameter (0.7) which regulates the randomness of the generated text, and the top-k parameter (40) which constrains the model's vocabulary during generation. Additionally, the song title is provided as a prefix to guide the generation process, ensuring that the generated lyrics are coherent and relevant to the specified theme. The generation is performed five times to produce multiple variations of the lyrics. The generated lyrics are returned as a list of text sequences.

Lyrics similarity

The generated text is concatenated with the existing Eminem lyrics dataset, forming a new corpus of lyrics for comparison. The combined lyrics corpus is transformed into TF-IDF vectors using scikit-learn. This vectorization step represents each lyric as a numerical feature vector. Cosine similarity is computed between the TF-IDF vectors of the generated lyric and all other lyrics in the corpus. Cosine similarity measures the similarity between two non-zero vectors in an inner product space. The top three most similar lyrics to the generated lyric are identified based on the computed cosine similarity scores. These similar lyrics provide insight into how closely the generated lyrics resemble existing Eminem lyrics. This information offers a comparative analysis of the generated lyrics and their similarity to known Eminem songs, aiding in evaluating the quality and authenticity of the generated content.

Experimental results

The output includes the index of the generated lyric, the titles of the most similar existing Eminem lyrics, the cosine similarity scores, and the Data Frame indices of the most similar lyrics. This information offers a comparative analysis of the generated lyrics and their similarity to known Eminem songs, aiding in evaluating the quality and authenticity of the generated content.



```
lyrics 0:
- similar to: Beautiful, Little Engine, The Ringer
- scores: 0.090, 0.088, 0.083
- df indices: [ 26 129   6]

lyrics 1:
- similar to: Big Weenie, Guts Over Fear, Drug Ballad
- scores: 0.120, 0.097, 0.092
- df indices: [199  38  90]

lyrics 2:
- similar to: Mic Check One Two, Jimmy Crack Corn, Discombobulated
- scores: 0.081, 0.080, 0.074
- df indices: [331 233 176]

lyrics 3:
- similar to: No Love, Kim, Em360 Rapcity Backroom Freestyle
- scores: 0.131, 0.120, 0.116
- df indices: [ 24  50 266]

lyrics 4:
- similar to: Remember Me?, I Remember (Dedication to Whitey Ford/Everlast Diss), Never Love Again
- scores: 0.231, 0.175, 0.113
- df indices: [131 239 135]
```

Fig. 2 – Similarity Analysis

The generated lyrics are subjected for similarity analysis. The lyric with the highest similarity percentage is the lyric which is more similar to Eminem's song existing's lyrics.

Conclusion

In this study, we proposed a model to generate lyrics using a novel approach. The model is trained with the help of a dataset containing lyrics created by the artist Eminem. Using this data, the model generates an understanding of word correlations and creates a pattern to formulate or recreate lyrics in a similar manner. After the training phase of the model, the model generates similar lyrics based on its training and knowledge. This newly generated lyrics is then compared to the original dataset to create an error function which further testifies its accuracy and efficiency. Based on these findings, we can conclude whether or not the model can generate accurate and comprehensible lyrics.

Acknowledgements

We would like to express our sincere gratitude to Viswajyothi College of Engineering and Technology, Vazhakulam for their generous support. We would like to express our sincere gratitude and heartfelt thanks to Dr. Anita Brigit Mathew, Mrs. Mary Nirmala George and Mrs Anupriya Mohan of the Artificial Intelligence and Data Science Department for their motivation, assistance, and help. We also thank all the faculty members of the Artificial Intelligence and Data Science Department for their constant support and encouragement.

References

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

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.

Gatt, A., and Krahmer, E. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research 61:65–170.

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.

Newman, D.; Lau, J. H.; Grieser, K.; and Baldwin, T. 2010. Automatic evaluation of topic coherence. In Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics, 100–108.