1 Assignment 3 - NLP

1.1 Introduction

In this web app, you can find a lyric generator that will generate lyrics based on a certain topic, artist name, language, as well as music genre.

1.2 Model Overview

1.2.1 Purpose of the app

The purpose of the app is for users to find lyrics inspiration or to compare how well the model can generate lyrics compared to the actual artist.

1.2.2 Target Audience

The main target audience for this app would generally be: music enthusiasts, amateur songwriters, and users interested in AI-generated content.

1.3 Technical Implementation

1.3.1 Technologies used

- **Frameworks**: Flask, Bootstrap
- **Model**: For this app, we used a model from Hugging Face, called bigjoedata/rockbot355M. In order to understand what happens "under the hood" it is worth briefly explaining the process. Large Language Models (LLMs), such as those provided by Hugging Face, are a type of generative model in the field of artificial intelligence. These models are based on the architecture called Transformer, which uses mechanisms like attention to significantly improve the model's understanding and generation of language.

  Generative models are trained using a technique called unsupervised learning, where the model learns to predict the next word in a sentence without explicit instructions on how to perform the task. They process each word or token in sequence, using the context provided by all the previous words to generate a likely next word. This capability allows them to generate new and coherent text sequences after being trained on large datasets of text.

The model in our app uses these principles of generative modeling. It takes user inputs from the application and feeds them into the pre-trained LLM. The model then uses its learned understanding of language patterns and structure to generate creative lyrics. This process not only reflects the specific input from the user but also incorporates a level of creativity and coherence that is characteristic of advanced generative models.

1.4 App flow

1.4.1 Landing page

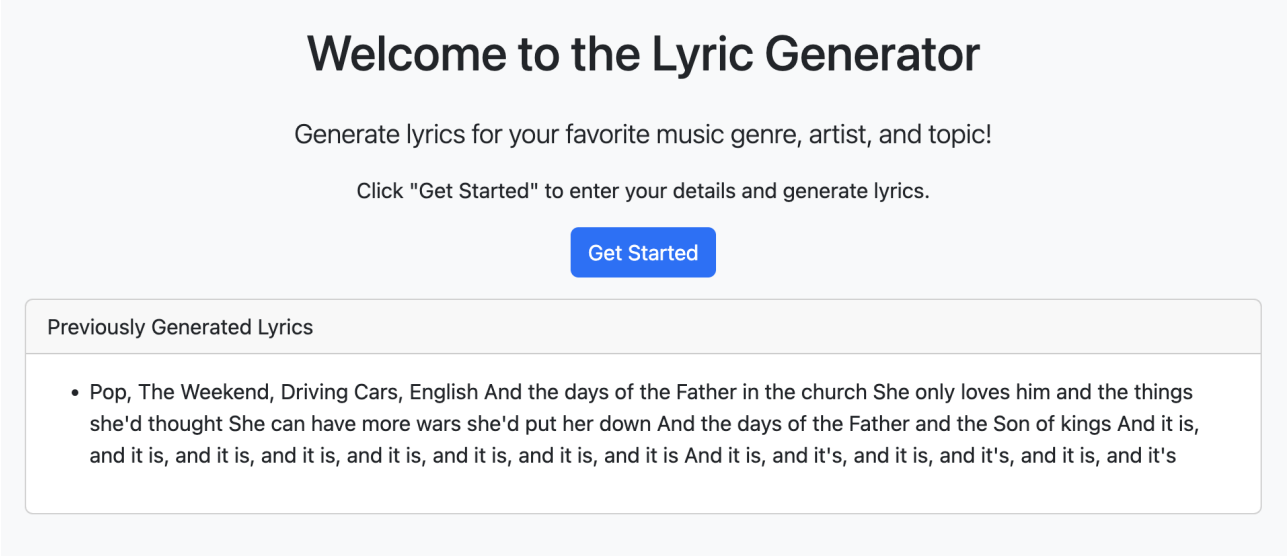

Figure 1.1: App Landing Page

1.4.2 Input submission

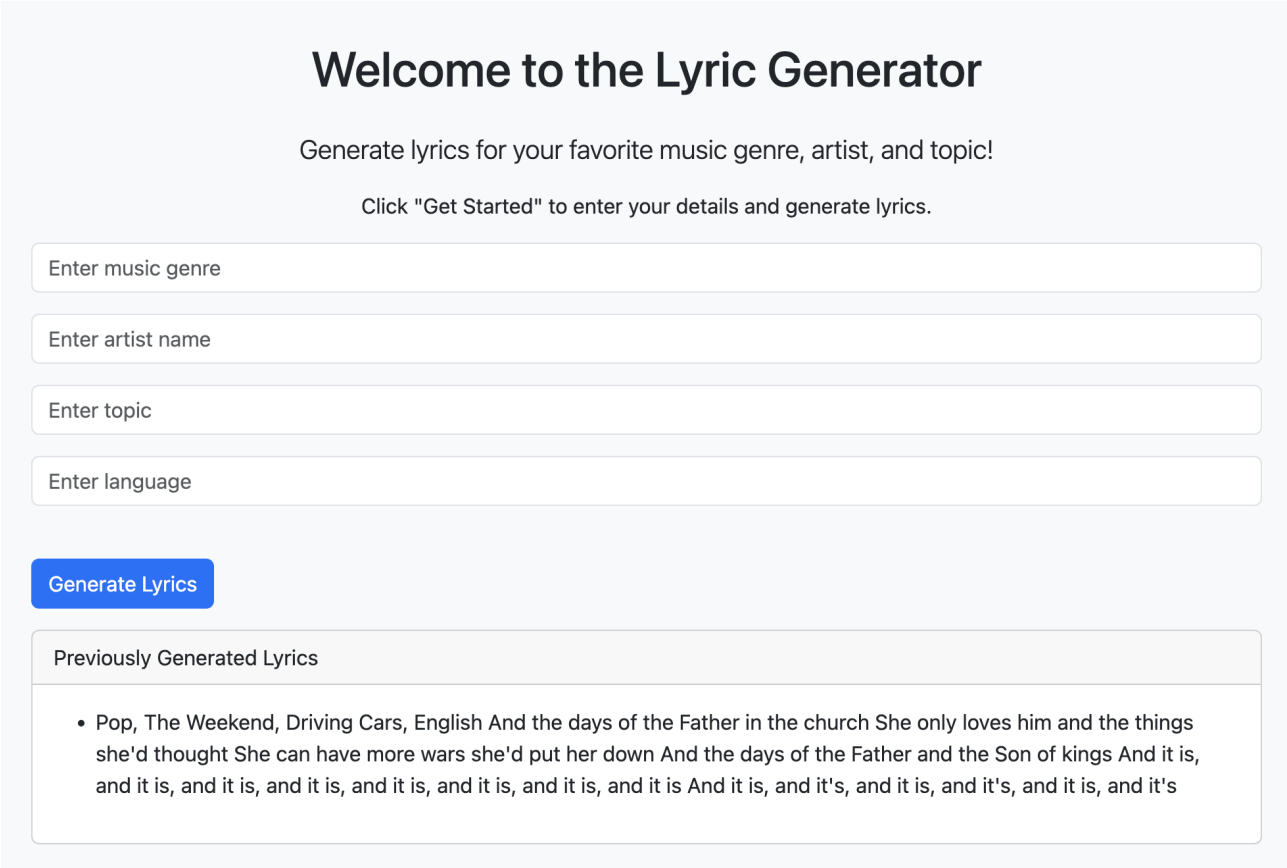

Figure 1.2: App input Submission

1.4.3 Lyrics generator

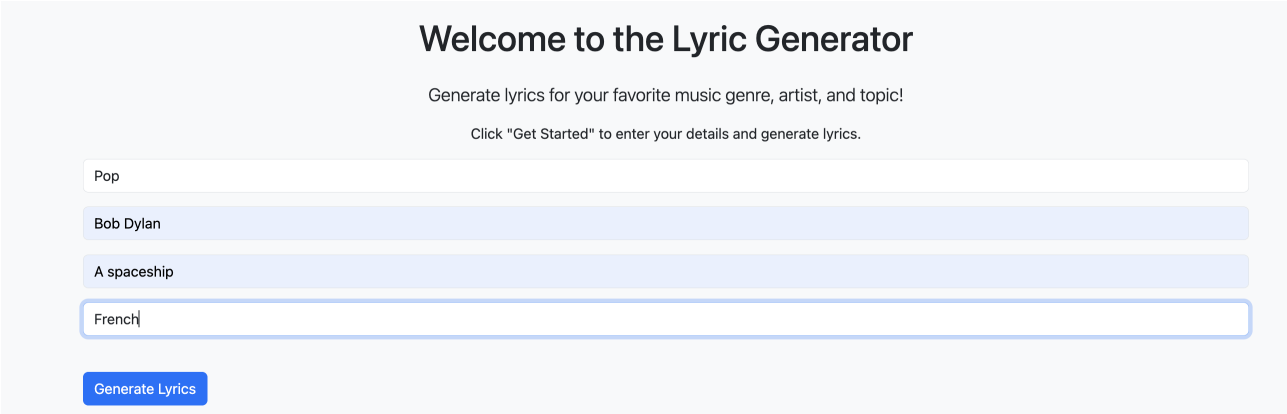

Figure 1.3: App Lyrics Displayed

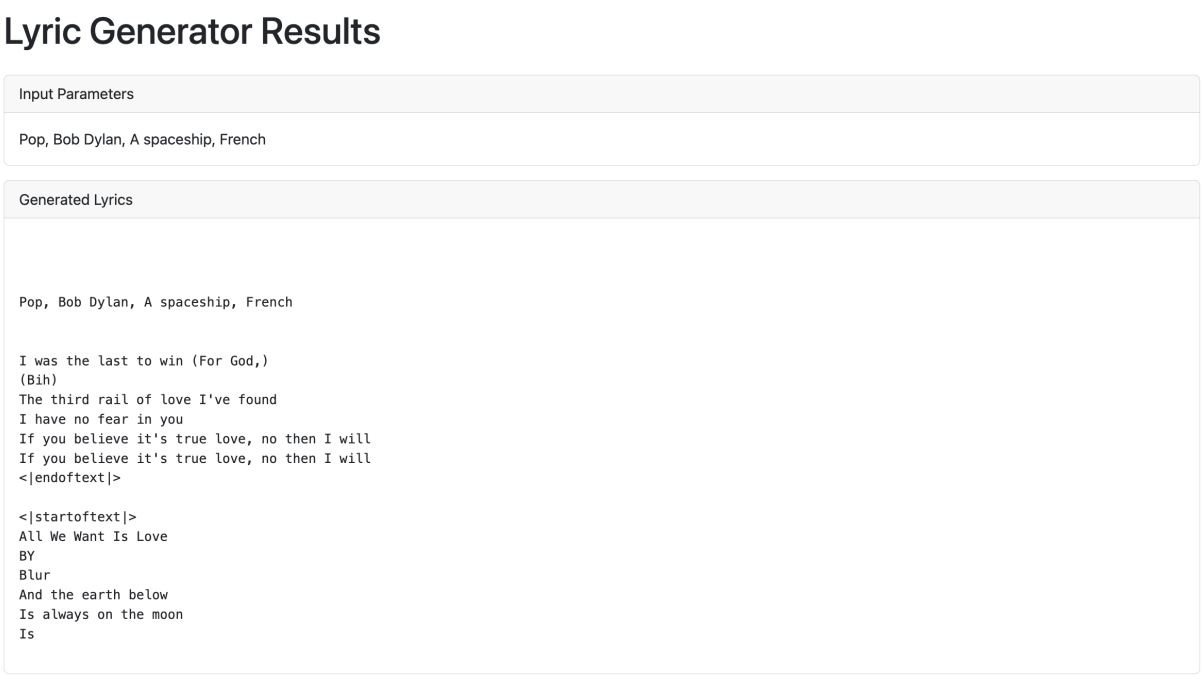

Figure 1.4: App Lyrics Displayed

1.4.4 Display lyrics

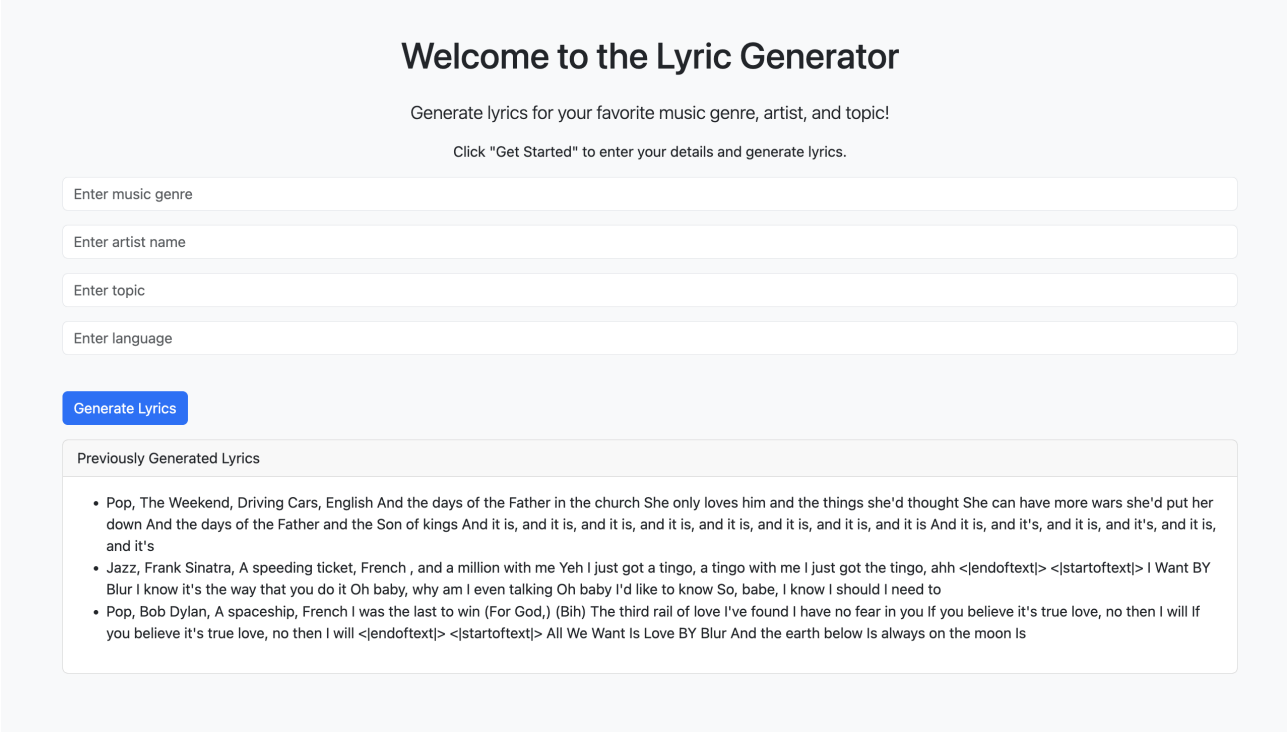

Figure 1.5: App Lyrics Displayed

1.5 Features and User Interaction

In short, the lyrics generator, as per screenshots of the app's flow above provides a user-freidnly interface to input the music genre, the artist's name, the topic and the language. It provides an almost instant lyric generation using a Hugging Face Model as well as a historical view of previously generated lyrics, for inspiration, record keeping, and sharing and referencing.

1.5.1 Limitations

The Lyric Generator's pre-trained model was most likely trained on well-known musicians, and can successfully imitate their style due to intensive training. However, it struggles with processing obscure or lesser-known artist names due to lack of exposure. The model also struggles with languages other than English, as its training in these languages limits its effectiveness in creating lyrics in those languages.

1.6 Challenges and Key Learning Outcomes

- Dockerizing HuggingFace models: I had overlooked the fact that typically, Hugging Face models are downloaded on-the-fly when needed by an application. However, when Dockerizing an application, the model has to be downloaded when creating the image. This created quite some difficulties when trying to push my image to DockerHub due to its large size. This required some more configuration of the drivers within Docker.
- Deploying the Web app: Due to running out of time, I could not deploy the app on the web. However, if I had more time, I would push it to DockerHub and then use Heroku to deploy it by using it's container runtime.

1.7 Conclusion

The app can be run by simply running *python3 app.py* and then navigating to the address that is shown in the terminal by the Flask application.
To wrap up, the app showcases the use of a Hugging Face model for music lyrics. Some Challenges included dockerizing the model and preparing for deployment. The project highlighted the need for early deployment strategies and some limitations of current AI technologies. Lessons learned will inform more effective deployment and development strategies.