Report: Song Lyrics Generation Using Neural Networks

Introduction

The goal of this project is to generate song lyrics based on the artist's name and genre using a dataset of 57,650 Spotify songs. The project is divided into three milestones. This report covers Milestone 2 and Milestone 3, detailing the methodology followed, model architecture, training, evaluation results, and findings.

Milestone 2: Building a Neural Network Model from Scratch

Methodology

Data Preprocessing:

1. Loading and Cleaning Data:

- The dataset was loaded using Pandas.

- Missing values and duplicate entries were removed to ensure data quality.

2. Text Tokenization: - The NLTK library was used to tokenize the lyrics, removing stop words and non-alp

3. Creating Sequences: - The text data was converted into sequences of tokens using TensorFlow's Token

Model Architecture:

The neural network was built using TensorFlow and Keras. The architecture includes:

- Embedding Layer: Converts input tokens into dense vectors of fixed size.

- LSTM Layers: Two bidirectional LSTM layers with 256 units each to capture long-term dependencies in the text.

- Dropout Layers: Dropout layers with a rate of 0.5 to prevent overfitting.

- Dense Layer: A dense layer with softmax activation to predict the next word in the sequence.

Training:

- The model was trained using a subset of the dataset.

- Early stopping and model checkpointing were used to monitor the validation loss and save the best model.

- The model was trained for 5 epochs with a batch size of 128.

Evaluation:

- The model was evaluated on the test set to determine the loss and accuracy.

- Although the primary evaluation criterion was the architecture, the model achieved a test accuracy of 12.78%.

Milestone 3: Fine-Tuning a Pre-trained Model

Methodology

Loading Pre-trained Model:

- The model trained in Milestone 2 was loaded for fine-tuning.

Data Preprocessing:

- Similar to Milestone 2, the dataset was processed in chunks to handle large data efficiently.

Data Generator:

To handle large datasets efficiently and avoid memory issues, we used a data generator for processing data in batches during training.

Fine-Tuning:

- The model was fine-tuned on the entire dataset using the data generator to manage memory usage.

- The training process involved additional epochs, leveraging early stopping and learning rate reduction.

Evaluation:

- The fine-tuned model was evaluated on the test set, showing improved performance.

- The model achieved a test accuracy of 15.32% after fine-tuning.

Text Generation:

- The fine-tuned model was used to generate lyrics based on a seed text.

Findings and Conclusions

Findings:

- The model architecture with bidirectional LSTM layers and dropout proved effective in capturing

the context of lyrics.

- Fine-tuning the pre-trained model on the entire dataset improved the performance, demonstrating the effectiveness of transfer learning.

- Using a data generator allowed us to handle large datasets efficiently, preventing memory issues and ensuring smooth training.

Conclusions:

- The approach of using a neural network model with LSTM layers for text generation in the context of song lyrics is promising.

- Fine-tuning a pre-trained model on a larger dataset can significantly enhance the model's performance.

- Further improvements can be achieved by experimenting with different architectures and hyperparameters, and by using more sophisticated text generation techniques.

Overall, the project demonstrates a successful application of deep learning techniques to generate song lyrics, providing insights into the potential of neural networks for creative text generation tasks.