https://github.com/tomilayoadetunji/Assignment-13---Generative-AI-Essentials/upload

Generative AI Text Generation Using LSTM Models: A Hands-On Implementation

1. Introduction

Generative Artificial Intelligence (Generative AI) represents a transformative class of machine learning systems that learn underlying patterns from data and use this knowledge to generate new content. These systems are essential in modern applications including conversational agents, content creation pipelines, data augmentation, and creative storytelling. Unlike discriminative models that classify or predict labels, generative models focus on producing entirely new sequences that resemble the training data.

This report explores generative modeling through a hands-on implementation in Google Colab. A Long Short-Term Memory (LSTM) neural network is trained on Lewis Carroll's 'Alice in Wonderland' to demonstrate how sequence models learn linguistic patterns and produce coherent outputs.

2. GPT Architecture Overview

Modern GPT models use a Transformer decoder architecture characterized by:

- Self-attention mechanisms enabling long-range context modeling.

- Positional encodings for order awareness.

- Stacked decoder blocks allowing deep contextual learning.

- Autoregressive training predicting the next token given prior tokens.

- Tokenization to convert text into numerical inputs.

GPTs generate text by sampling from learned probability distributions, updating their context with each predicted token.

3. Methodology

3.1 Dataset Preparation

A public-domain literary text, 'Alice in Wonderland,' was downloaded from Project Gutenberg. The dataset contained over 143,000 characters and more than 3,000 unique words. The text was lowercased and stripped of unnecessary whitespace.

3.2 Tokenization and Sequence Creation

Using TensorFlow's Tokenizer, the text was converted into 3,067 unique tokens. Word sequences of length 10 were created, generating 27,755 training samples. Predictors (X) and labels (y) were created and one-hot encoded.

3.3 Model Architecture

A Keras LSTM model was built with:

- Embedding layer (64-dimensional embeddings)
- LSTM layer with 128 units
- Dense softmax output over the vocabulary

The model was compiled using Adam (learning rate 0.001) and trained for 10 epochs.

4. Results and Practical Demonstration

4.1 Text Generation

A text generator function was implemented using tokenization, padding, probability sampling, and temperature scaling. A seed phrase—"alice was beginning to feel very tired"—was extended into a 30-word generated passage.

4.2 Content Creation Application

A story-intro generator demonstrated an example application. Given a user prompt such as "in a strange land of talking rabbits," the model produced a whimsical narrative introduction modeled after the style of the dataset.

5. Ethical Considerations

Generative AI raises ethical concerns including:

- Bias propagation from training data
- Risk of misinformation
- Copyright and data-ownership issues
- Hallucination of false information

Mitigation strategies include dataset filtering, human oversight, safety alignment, and transparent documentation.

6. Conclusion and Future Directions

This project demonstrates a complete workflow for building a generative text model using LSTM networks—from data preparation to text generation. While the model produced coherent short sequences, future improvements may include:

- Using Transformer-based architectures

- Training on larger datasets

- Implementing subword tokenization

- Applying advanced decoding strategies such as beam search