Word-Level Language Model

Overview

This project implements a word-level language model using PyTorch, designed to predict the next word in a sequence based on a given context. The model employs a transformer-based architecture with multi-head self-attention, suitable for natural language processing tasks such as text generation.

Features

- **Transformer Architecture**: Includes token and positional embeddings, multi-head self-attention, feed-forward layers, and layer normalization.
- **Word-Level Tokenization**: Processes text at the word level using regular expressions for tokenization.
- **Dynamic Vocabulary**: Builds a vocabulary from input text files, including an <UNK> token for out-of-vocabulary words.
- **Training and Evaluation**: Supports training with AdamW optimizer and evaluates train/validation losses periodically.
- **Text Generation**: Generates coherent word sequences by sampling from predicted probabilities.

Requirements

- Python 3.8+
- PyTorch
- Matplotlib (optional, for plotting)
- A folder containing .txt files for training data

Usage

1. Prepare a folder with .txt files containing the training text.
2. Update the folder_path variable in the script to point to your text files directory.
3. Run the script:
4. The model will train for 500 iterations, print train/validation losses, and generate a sample text sequence.