Project Documentation

Overview

This project involves developing a system to scrape LinkedIn posts, preprocess the data, fine-tune a language model (GPT-2 or GPT-Neo), develop a user interface for generating text based on user input, and evaluate the model's performance. The aim is to generate text that mimics the style of specific LinkedIn posts.

1. Scraper and Preprocessing

Setup Instructions

Install necessary Python packages: *selenium, webdriver_manager, pandas*.

Ensure ChromeDriver is installed and its path is set correctly.

How It Works

The script uses Selenium to automate web browsing, log into LinkedIn, and scrape posts.

Posts are collected into a DataFrame and saved as a CSV file for further processing.

Preprocessing includes tokenization and cleaning of the data, making it suitable for model training.

2. Trainer

Setup Instructions

Install the *transformers* and *torch* libraries from Hugging Face.

Load and fine-tune the GPT-2 or GPT-Neo model on the preprocessed data.

Training Process

The model is fine-tuned using the provided training script.

Training parameters such as learning rate, epochs, and batch size can be adjusted in the script.

The model is saved periodically during training to prevent loss of progress.

3. Interface

Setup Instructions

Install *flask* for the web interface.

Use the provided *interface-2.py* script to run the Flask server on GPT-2 model.

Using the Interface

The web interface allows users to enter prompts.

The system generates responses based on these prompts using the fine-tuned model.

4. Evaluation

Methodology

Manual Qualitative Evaluation: Assess the generated text for coherence, relevance, and style similarity.

Automated Quantitative Evaluation: Use BLEU scores to measure lexical similarity between generated text and reference posts.

Guide:

Generate text using a variety of prompts.

Compare the generated text to original posts for style and content.

Use the provided script to calculate BLEU scores for a quantitative assessment.

Download and add the "model.safetensors" to the "trained_model" and "trained_model_neo" folders from below link:
https://drive.google.com/drive/folders/1eI78-oQCSamD0xSfTJuHx4EAwJKat4mI?usp=sharing