Task: Predictive Text Model with ChatGPT Integration

file would be found in **:**

Development Environment:

- Google Colab: A cloud-based platform that provides free access to GPUs,

Dataset:

- Metamorphosis.txt: A novella written by Franz Kafka,

Natural Language Processing Library:

- NLTK (Natural Language Toolkit): Used for tokenization through the RegexpTokenizer to process and prepare the text data for training.

Machine Learning Libraries:

- TensorFlow with Keras: TensorFlow is the machine learning framework, and Keras is a high-level neural networks API that runs on top of TensorFlow.

Model Architecture:

Tokenization:

- NLTK RegexpTokenizer

Model Built using a two-layer LSTM (Long Short-Term Memory) architecture followed by a dense layer with softmax activation.

- LSTM
- Dense
- Softmax

Input Layer:

n_words: The number of words in each input sequence. i.e trained with 10 and 12 n_words.

len(unique_tokens): The number of unique tokens in the training data. i.e. 2572

LSTM Layer 1:

- Units: 128
- Input shape: (n_words, len(unique_tokens))
- Return sequences: True (to provide the full sequence output for the next LSTM layer)

LSTM Layer 2:

- Units: 128

Dense Layer:

- Units: len(unique_tokens)
- Activation: Softmax

Training Parameters:

- Optimizer: RMSprop

- Learning Rate: 0.01
- Loss Function: Categorical Crossentropy
- Batch Size: 128
- Epochs: 20
- Shuffle: True (to shuffle the training data in each epoch)

Key aspect of the Training:

- Train Loss : 1.07
- Train Accuracy: 79.99%
- Test Loss: 7.72
- Test Accuracy : 8.8%

The large difference between the training and test loss/accuracy suggests overfitting. The model seems to have memorized the training set but does not generalize well to new or unseen data.

ChatGPT integration

Python script that combines the predictive text model trained and the OpenAI GPT-3.5-turbo model for a more interactive and context-aware sentence completion.

Requirement

- openAI package
- openai API Key

WorkFlow

1. Load Predictive Text Model.
2. Load Token Index.
3. Function to Predict the next word.
4. Main Interaction Loop.
5. Generate Complete Sentence using chatGPT model.

Instruction of use:

Download:

- modelv3.h5
- unique_token_index.json
- gpt100.py

All files in the same directory.

Requirements:

- python 3
- openai
- tensorflow.pytorch
- openai API KEY
    - Put the openai api key in .env file with keyword “API_KEY”

Run : $ python gpt100.py

Challenges

- Limited Training Data
- Limited Hardware Resources
- OpenAI API Rate Limits
- Model Hyperparameter Tuning