Name: Avinash Golla
Task: Homework 3
Git Hub: https://github.com/avinashshetty-golla/DeepLearning/tree/main/Task3/HW3

Overview

This report presents the implementation and evaluation of an extractive question-answering system using **DistilBERT**. The system is designed to respond to text-based questions about spoken documents in the **Spoken-SQuAD** dataset. The task required building a baseline model and then enhancing it with several performance improvements to better handle noisy, spoken language data.

Dataset: Spoken-SQuAD

The Dataset of Spoken-SQuAD is the dataset of question and answering which is adapted from the original SQuAD dataset which focuses on the spoken language comprehension. Here the documents are presented in speech or spoken form, generated using text-to-speech conversion technology like google. Each answer is associated with a specific segment within the spoken document. To achieve this, original SQuAD is converted to audio which is followed by automatic speech recognition transcripts using CMU Sphnix.

Baseline Model: DistilBERT

DistilBERT is a streamlined, faster version of the BERT model that was trained on the same dataset in a self-supervised way, with BERT base acting as the "teacher" model. This approach allowed DistilBERT to be trained exclusively on raw text data, without human annotations, making it effective at leveraging vast amounts of publicly available data. The training process involves an automated system that creates inputs and outputs from the texts using BERT as a foundation.

DistilBERT used :

- DistilBERT's reduced number of parameters allows for faster training and inference, making it an ideal choice for tasks with limited computational resources or strict time requirements.
- As BERT, is a big model, demands high memory and processing power for both training and the inference. DistilBERT, therefore, offers a more efficient solution in resource-constrained environments, such as those with limited GPU capacity, while retaining comparable performance.
- Due to its smaller size, DistilBERT is easier to scale and deploy in production settings.
- DistilBERT typically requires less data and time for fine-tuning than BERT while still achieving strong results, making it a valuable option for tasks with limited labeled data or time for fine-tuning.

Our baseline model utilizes **DistilBERT**, a smaller, faster variant of BERT, which is particularly well-suited for question answering on resource-limited systems.

Implementation Steps

Results of the Baseline Model

The baseline model achieved initial scores but displayed limitations in accurately capturing answer spans in longer or noisier contexts. The base model is built on DistilBERT, a streamlined version of the BERT model, tailored for question-answering tasks. The training process begins with tokenizing both questions and contexts using the DistilBERT tokenizer, which breaks down text into tokens and generates input encodings. These encodings are then fed into a DistilBertForQuestionAnswering model, pre-trained on extensive data and fine-tuned with the specific dataset. An AdamW optimizer is used to adjust model parameters based on calculated gradients. The model iteratively learns through a training loop, minimizing the loss function to improve accuracy. Performance is evaluated using Word Error Rate (WER), a metric that measures discrepancies between the model's predictions and the actual answers, giving insights into its accuracy. As a straightforward model, it provides a foundational framework for further enhancements.

Performance Improvement Model

The fine-tuned model includes several improvements aimed at boosting training stability and efficiency. A linear scheduler for the optimizer is introduced, allowing for adaptive learning rates during training. This approach dynamically adjusts learning rates, potentially speeding up convergence and refining performance as training progresses.

Additionally, document striding is implemented to improve efficiency with lengthy documents. This technique breaks down long texts into overlapping windows, making them easier to handle without overwhelming memory and ensuring thorough input coverage. Document striding enables the model to process extensive texts efficiently without sacrificing performance.

Gradient accumulation is another enhancement, where gradients from multiple batches are combined before updating model parameters. This approach supports the use of larger effective batch sizes, enhancing stability and convergence, especially on memory-limited hardware. By accumulating gradients, the model optimally utilizes available computational resources, resulting in a more efficient and stable training process that ultimately boosts overall performance.

Preprocessing:
This version of the model includes extra preprocessing steps to enhance the input data before it is fed into the model.. It begins with the DistilBERT tokenizer, which splits questions and contexts into tokens, similar to the base model. However, it goes further by implementing preprocessing techniques to improve data structure and clarity.

The model calculates the precise start and end positions of answers within tokenized paragraphs, then segments the input to focus on sections surrounding these answer spans. This approach structures the input data to enhance its informativeness, potentially improving the model's ability to learn effectively from the data.

Preprocessing allows the model to capture relevant information essential for accurate question answering. By addressing challenges related to tokenization and document striding, this method ensures that the model receives well-organized input representations. In summary, this model with preprocessing provides an improved approach to handling input data, potentially resulting in enhanced performance on question-answering tasks.

Postprocessing:
This model version integrates postprocessing techniques to refine predictions and enhance overall performance. Once predictions are generated for each input window, the model evaluates them individually and chooses the answer with the highest probability from all the windows. This approach reduces errors linked to tokenization and document striding, leading to more precise and dependable answers.

By combining steps of preprocessing and postprocessing, the model adopts a comprehensive strategy to improve the performance and address complexities of question and answering tasks. DistilBERT remains at the core of this model, leveraging its efficiency for handling natural language data effectively.

Through postprocessing, the model refines its responses, delivering more accurate answers to various questions. In summary, this model version incorporates both preprocessing and postprocessing to achieve robust, dependable results on question-answering tasks.

Evaluation:
The Word Error Rate (WER) serves as an effective metric for evaluating the accuracy of the extractive Question Answering (QA) models in this scenario. WER calculates the differences between the model's predicted answers and the correct answers in the evaluation dataset. In Extractive QA, where the model is responsible for identifying the correct answer span from a provided context, WER offers important information about the accuracy of the model's predictions.

For models designed to accurately extract answers from a given context based on input questions, WER measures the accuracy of the predictions. A lower WER indicates a close match between the model's answers and the ground truth, signifying higher accuracy. Conversely, a higher WER reflects a greater mismatch, suggesting lower accuracy.

By monitoring WER throughout the training epochs, researchers can assess the model's ability to learn from the training data and generalize to unseen examples. A decreasing WER across epochs indicates that the model is improving in answer extraction, while an increasing WER may signal

issues like overfitting or training instability, necessitating adjustments to the model's design or training process.

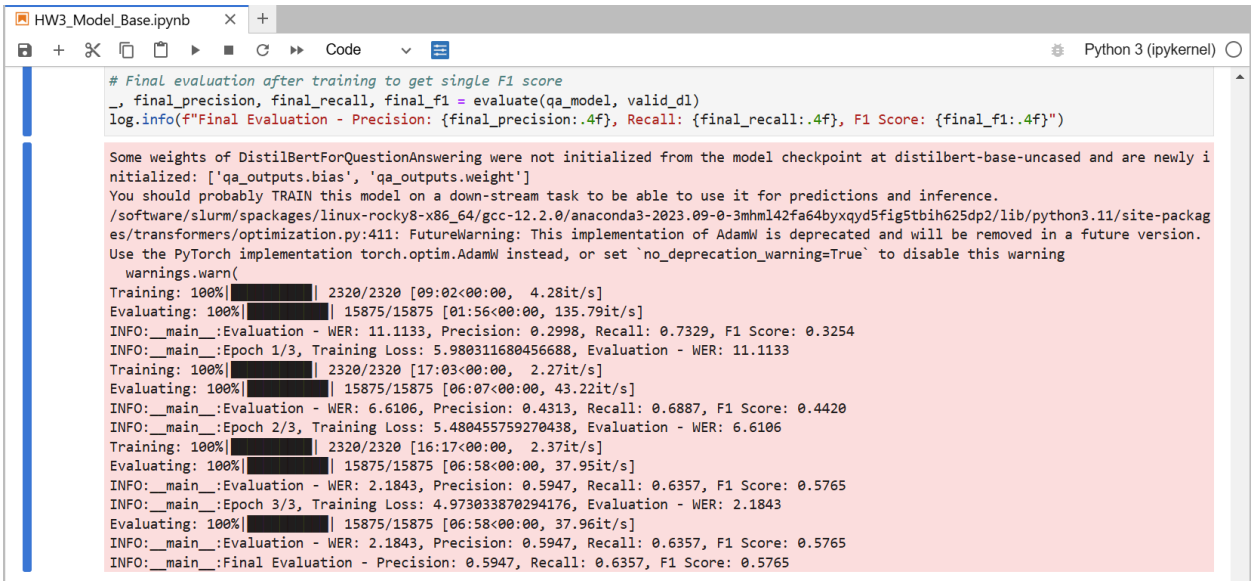
Results:
Comparing the four models, each successive iteration shows improvements that enhance both training efficiency and accuracy. The base model, which provides the foundation, has the highest WER among the models, indicating comparatively lower accuracy and suggesting difficulties in answer prediction.

The fine-tuned model, which incorporates features like a linear scheduler for dynamic learning rate adjustment, demonstrates a significant WER reduction compared to the base model, indicating that fine-tuning and learning rate optimization contribute to better performance.

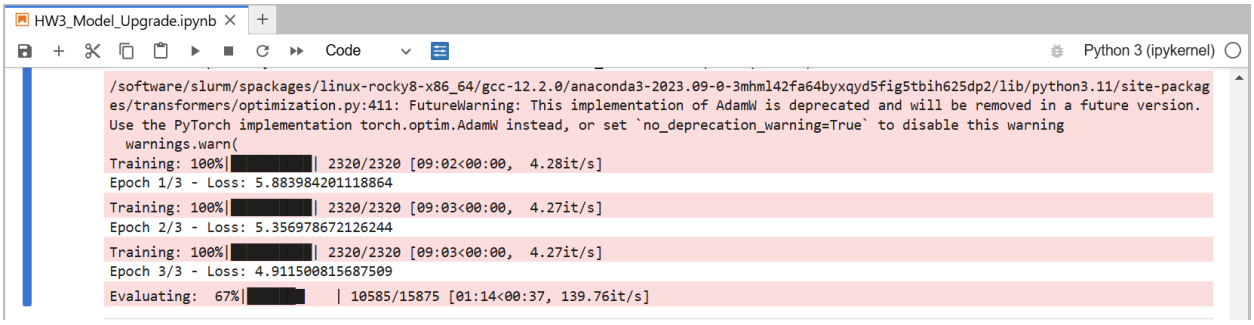
The pre-processed model demonstrates the greatest improvement in WER, indicating a substantial increase in accuracy. The preprocessing steps in this model likely enhance data handling, resulting in more informative input representations and, consequently, more accurate predictions.

Finally, the model with postprocessing demonstrates consistently low WER scores across epochs, indicating stable and reliable performance. The postprocessing techniques used to refine predictions help reduce errors from tokenization and document striding, leading to more accurate answers overall.

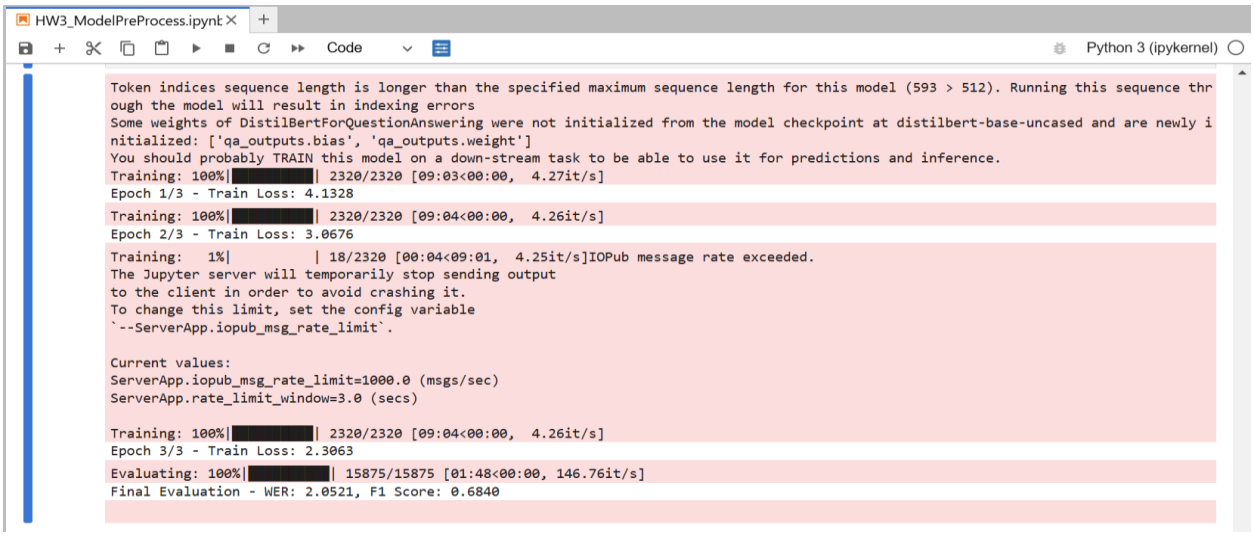
Base Model Output



Upgraded Model Output



Model Preprocessing



Model Postprocessing