CPSC-8430: Deep Learning - Homework 3

The main aim of this assignment is to build an extraction-based QA model that identifies the span of a text in a paragraph answering a given question. We identify the starting, ending positions of this span, and return the answer from the paragraph itself.

Github Code Link: https://github.com/laxman2405/DeepLearning_HW3/tree/main

Introduction

For this assignment, I worked on two popular transformer-based models BERT and DistilBERT. Both of them are developed by Hugging face and these models can be of great use for building question answer models. For training purposes, I have chosen the given data set Spoken Question Answering Dataset (SQuAD) which consists of 37,111 question answer pairs as training set and 5,351 as the testing set.

Data Processing and Fine-Tuning

The extraction, transformation and evaluation of the model is done in a separate file for each of the improvements we made. This file has the below components:

1. SpokenSquad dataset class to manage the encoding.
2. A custom QAModel initialized with a BERT/DistilBERT based model based on invocation.
3. A function to process data from train, test json files and outputs start, end positions and dictionary of encodings.
4. Also, normalizes and cleans the text to compute the F1 score.
5. We also have a function to calculate the loss function using focal loss and evaluate the model to calculate the F1 scores and WER scores for predicted and actual answers.

Common Training Parameters - Chosen a learning rate of 2e-5, weight decay of 0.02, batch size of 16, 6 epochs, AdamW optimizer and sequence length of 512 tokens.

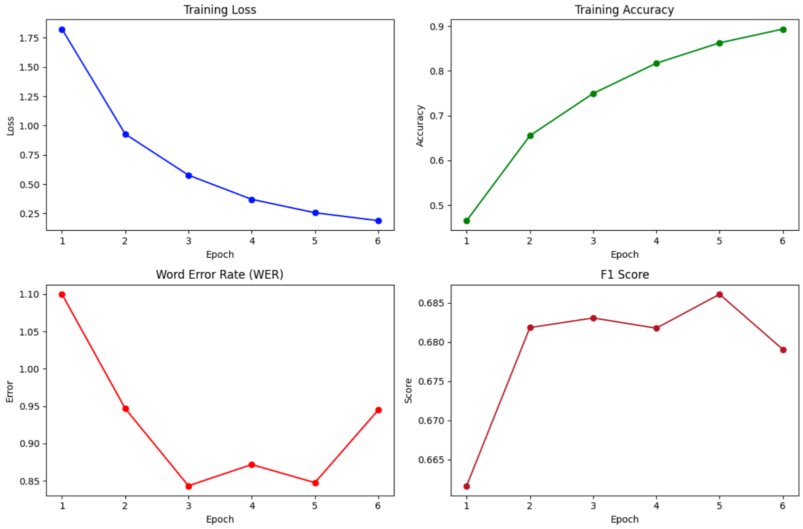
Pre-trained Models to avoid training phase and directly proceed with testing and evaluation can be found here:
https://drive.google.com/drive/folders/1vu6JyF8umbNHIuFbAnWs3igopr-zfPtW?usp=sharing

BERT Model

Based on transformer architecture, this model will process the text from both the directions and understand the context based on its preceding and following words, which leads to better understanding of the language. Below outlines each improvement, followed by its results across clean, Noise V1 (44% WER) and Noise V2 (54% WER) test conditions.

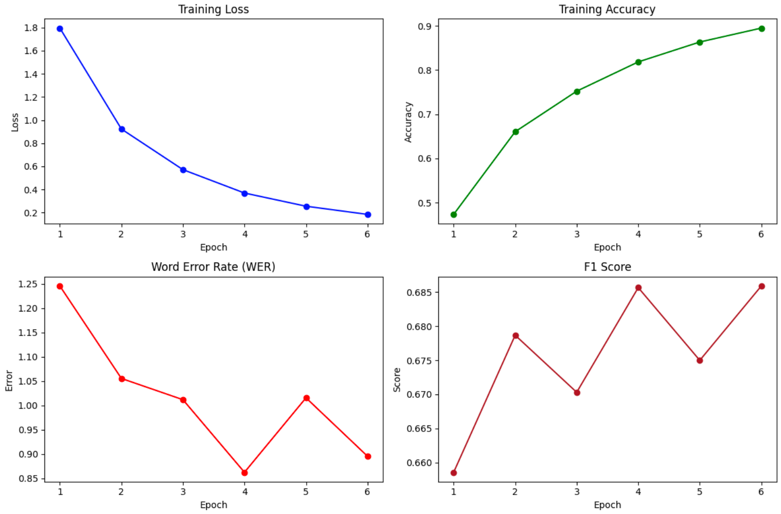
1. Simple Model

- This is configured to use **bert-base-uncased** with no additional improvements and trained over multiple epochs to find model's performance in terms of F1 score, training accuracy and word error rate.
- Results: It attained a F1 score of **0.67**, a training accuracy of **89.3%**, and WER of **0.94** on clean data. Under Noisy conditions, the performance declined with F1/WER scores of **0.39/2.50** for Noise V1 and F1/WER scores of **0.30/3.26** for Noise V2.



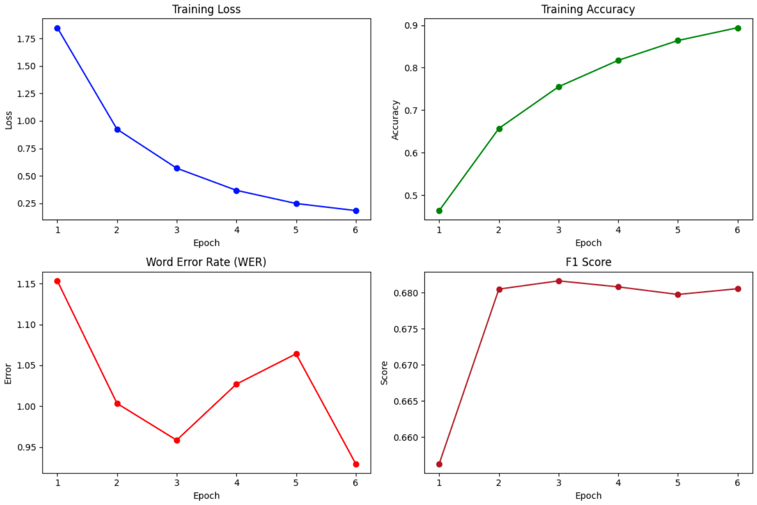
2. DocStride 128

- Uses the same model as base model, but we also add an additional parameter **stride = 128** in the tokenizer call while finding start, end positions. This change is used to manage large paragraphs, especially when the maximum length of the model exceeds.
- Results: A F1 score of **0.68**, a training accuracy of **89.4%**, and WER of **0.89** on clean data. Under Noisy conditions, the performance is F1/WER scores of **0.40/2.40** for Noise V1 and F1/WER scores of **0.32/3.63** for Noise V2.



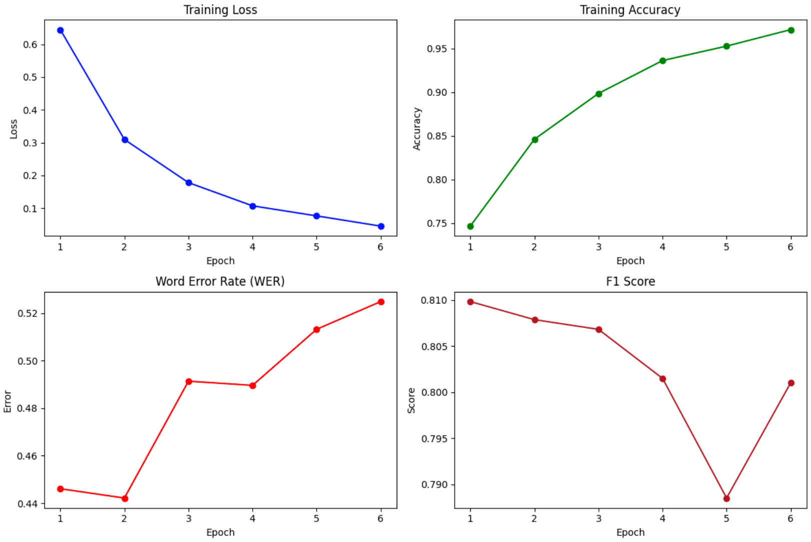
3. Exponential Learning Rate Scheduler

- A scheduler has been added along with docstride configuration to adjust the learning rate and this helps the model achieve better performance as it fine-tunes the dataset.
- Results: A F1 score of **0.68**, a training accuracy of **89.3%**, and WER of **0.92** on clean data. Under Noisy conditions, the performance is F1/WER scores of **0.40/2.36** for Noise V1 and F1/WER scores of **0.31/3.30** for Noise V2.



4. Enhanced Pre-trained Model

- Advanced model **bert-large-uncased-whole-world-masking-finetuned-squad** is used here along with above 3 improvements, as it has a larger capacity to understand more complex language patterns, giving more effective results.
- Results: An increase in F1 score of **0.80**, a training accuracy of **97.1%**, and WER of **0.52** on clean data. Under Noisy conditions, the performance is F1/WER scores of **0.48/1.88** for Noise V1 and F1/WER scores of **0.40/2.71** for Noise V2.



Observations from BERT models

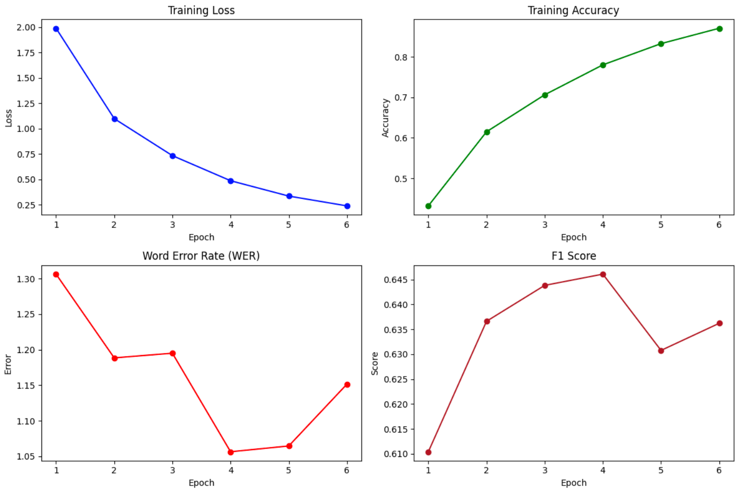
- If we observe, the base version of the model achieved an initial F1 score of **0.67** and accuracy of **89.3%**, but there is a slight drop in performance under noise conditions.
- Using the docstride method not only slightly improved F1 score and accuracy, but also performed better under noise conditions with improved F1 score of **0.40.**
- With the use of exponential LR scheduler with docstride, F1 score maintained at **0.68** on clean data, and there is a marginal improvement under Noise V1 with very less WER of **2.36.**
- A substantial improvement yields with the pre-trained model with F1 score of **0.80** and accuracy of **97.1%** and also it shows better resilience to noise as well.
- Overall, a sophisticated pre-trained model brings in most accuracy of all other models (base, docstride, scheduler), indicating the importance of a robust pre-trained model in cases involving clean data and noise.

DistilBERT Model

This is a compressed version of the BERT model and it is trained based on a technique called Knowledge distillation. It is usually useful for applications that need faster processing times, and also achieve almost 97% of BERT accuracy. Below outlines each improvement, followed by its results across clean, Noise V1 (44% WER) and Noise V2 (54% WER) test conditions.

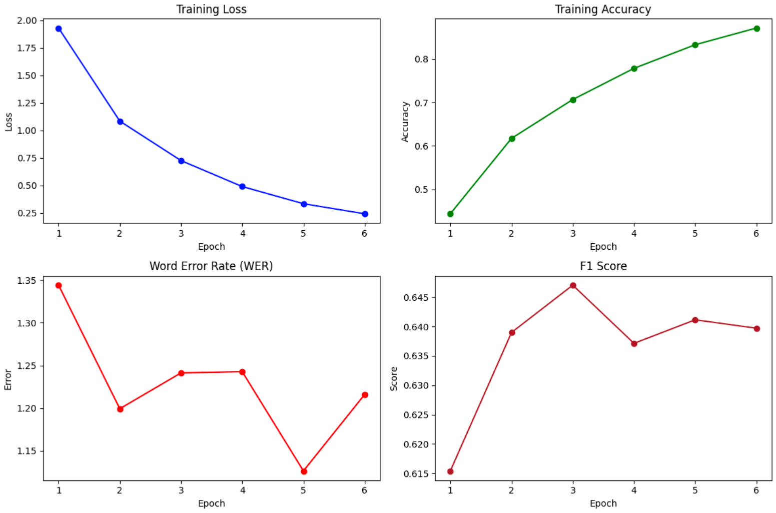
1. Simple Model

- This is configured to use **distilbert-base-uncased** with no additional improvements and trained over multiple epochs to find model's performance in terms of F1 score, training accuracy and word error rate.
- Results: It attained a F1 score of **0.63**, a training accuracy of **87.0%**, and WER of **1.15** on clean data. Under Noisy conditions, the performance is F1/WER scores of **0.37/2.97** for Noise V1 and F1/WER scores of **0.28/4.08** for Noise V2.



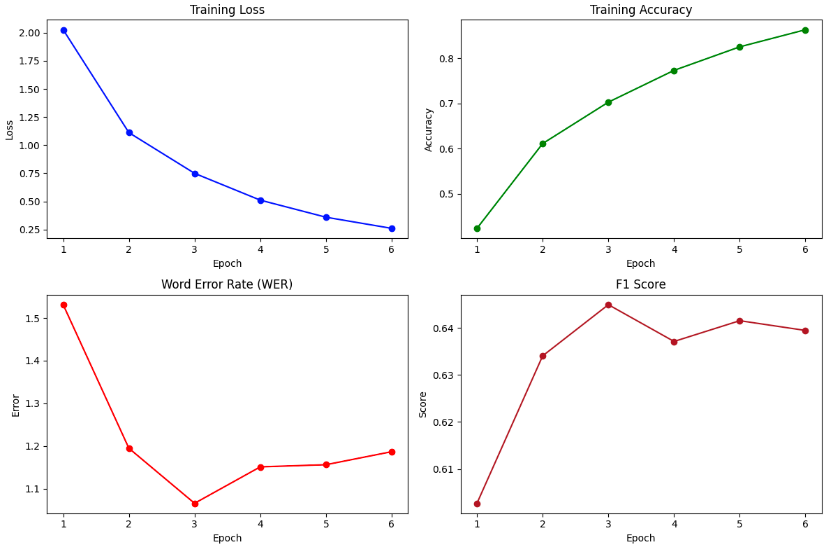
2. DocStride 128

- Uses the same model as base model, but we also add an additional parameter **stride = 128** in the tokenizer call while finding start, end positions. This change is used to manage large paragraphs, especially when the maximum length of the model exceeds.
- Results: A F1 score of **0.63**, a training accuracy of **87.1%**, and WER of **1.21** on clean data. Under Noisy conditions, the performance is F1/WER scores of **0.37/2.80** for Noise V1 and F1/WER scores of **0.29/4.21** for Noise V2.



3. Exponential Learning Rate Scheduler

- A scheduler has been added along with docstride configuration to adjust the learning rate and this helps the model achieve better performance as it fine-tunes the dataset.
- Results: A F1 score of **0.63**, a training accuracy of **86.31%**, and WER of **1.18** on clean data. Under Noisy conditions, the performance is F1/WER scores of **0.39/2.90** for Noise V1 and F1/WER scores of **0.31/4.08** for Noise V2.



Observations from DistilBERT models

- Achieved a good F1 score of **0.63** and accuracy of **87%** with distilbert base model, and with an improved addition of docstride slightly improved the accuracy to **87.1%.**
- Though a slight decrease in accuracy with scheduler, F1 score is still maintained. Also, this improvement performs well under noise conditions with increased F1 score compared to docstride and simple model.

Overall Comments

- From the observations, we can conclude that the bert based model, when enhanced with pre-trained model and additional improvements (docstride and scheduler) outperforms distilbert-base model.
- Thus, investing in larger pre-trained models to attain a high accuracy is essential. This approach will likely give us accurate results.