# HOMEWORK 3 (CPSC 8430 – DEEP LEARNING)

**SHRAVANI KONDA**

## Extractive Question Answering

The aim of this assignment is to develop a model which takes text paragraph and question as inputs and returns the answer which is in the paragraph.

## GitHub Link:

https://github.com/shravanik31/Deep-Learning/tree/main/HW3

## Dataset:

The dataset used is the Spoken Question Answering Dataset (SQuAD), that contains 37,111 records. Every record contains a paragraph providing context from which questions are drawn and to which the answers pertain, with single paragraph having multiple questions.

## Data Preprocessing:

The data preprocessing step utilizes tokenization, a fundamental technique where text is converted into tokens that can be fed into a model. I have used the DistilBertTokenizerFast from the pre-trained distilbert-base-uncased model, to extract and tokenize contexts, questions, and answers. The answers within the context are also aligned, establishing start and end positions for each token, which is crucial for training the model. The texts are also normalized to standardize answers for more consistent evaluation.

## Model:

I have used the **distilbert-base-uncased** model from Hugging Face which is smaller and faster version of BERT. DistilBERT is a transformer-based model pre-trained on a vast corpus of English text. It uses a self-supervised learning approach but is trained only on the task of Masked Language Modeling (MLM). The model consists of 66 million parameters and offers faster performance.

## Training:

Training parameters:

- Epochs: 6
- Learning rate: 2e-5
- Optimizer: Adam
- Loss function: Focal Loss

Throughout the training process, I tracked and recorded the loss and accuracy, comparing the model's predictions to the actual positions. The training concludes by calculating the average loss and accuracy across all batches and saving the model and tokenizer.

## Testing:

F1 and Word Error Rate (WER) scores are used to assess model's performance on a test dataset. To test the model, please either train the model by running the respective scripts or please download the contents of the respective pretrained models from the below link and evaluate by giving your path to the train and test datasets in the code.

https://drive.google.com/drive/folders/1sUOgLMbHMRywD9z7zNbwUufMfnBTSZWt?usp=sharing

## Results:

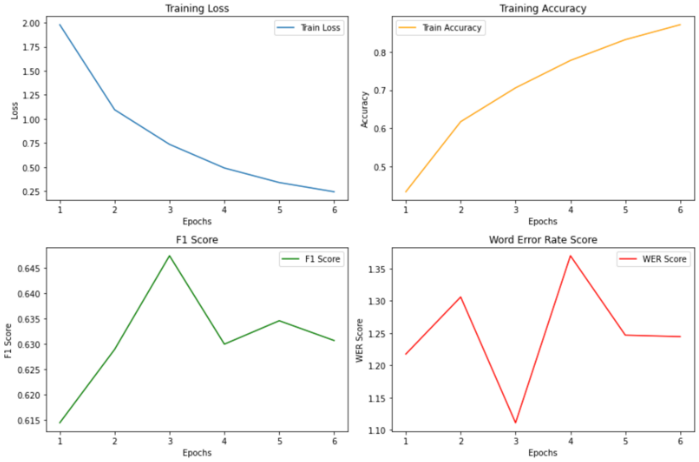
### Base model:

The base model (**distilbert-base-uncased**) is trained and evaluated on three testing datasets: No noise (WER (%) 22.73), Noise V1 (WER (%) 44.22) and Noise V2 (WER (%) 54.82).

Train Accuracy (No Noise): 0.8710
Train Loss (No Noise): 0.2426

| | No noise | Noise V1 | Noise V2 |
|---|---|---|---|
| F1 Score | 0.6306 | 0.3605 | 0.2708 |
| Word Error Rate | 1.2447 | 2.6860 | 3.9035 |

Below is the plot of the base model with no noise depicting Train Accuracy, Train Loss, F1 Score and WER over epochs.



## Improvements:

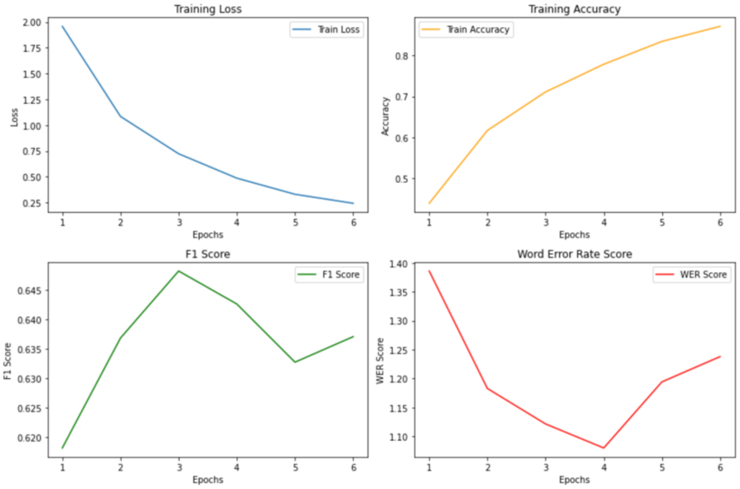
### Doc stride:

The doc_stride parameter ensures overlap between consecutive chunks of a long document, preventing answers from being split across chunks and improves the model's ability to accurately locate answers. I have used the doc_stride value as 128 in the base model.

Train Accuracy (No Noise): 0.8708
Train Loss (No Noise): 0.2439

| | No noise | Noise V1 | Noise V2 |
|---|---|---|---|
| F1 Score | 0.6370 | 0.3766 | 0.3043 |
| Word Error Rate | 1.2380 | 2.8778 | 4.1161 |

Below is the plot of the base model (no noise) with doc stride depicting Train Accuracy, Train Loss, F1 Score and WER over epochs.



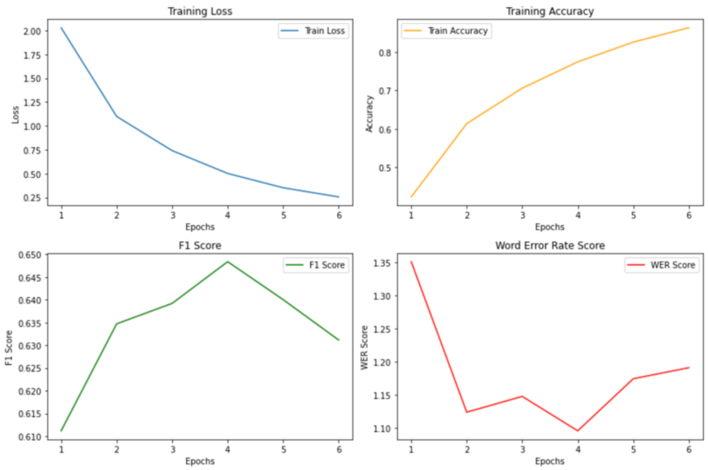
**Learning rate decay:**

By gradually reducing the learning rate using ExponentialLR as a scheduler (in my case - exponential rate of 2e-2), the model can make finer adjustments to its parameters as it gets closer to the optimal solution. Base model is tested along with doc stride and scheduler.

Train Accuracy (No Noise): 0.8630
Train Loss (No Noise): 0.2550

| | No noise | Noise V1 | Noise V2 |
|---|---|---|---|
| F1 Score | 0.6312 | 0.3850 | 0.3115 |
| Word Error Rate | 1.1909 | 2.7337 | 3.5013 |

Below is the plot of the base model (no noise) with doc stride, scheduler depicting Train Accuracy, Train Loss, F1 Score and WER over epochs.

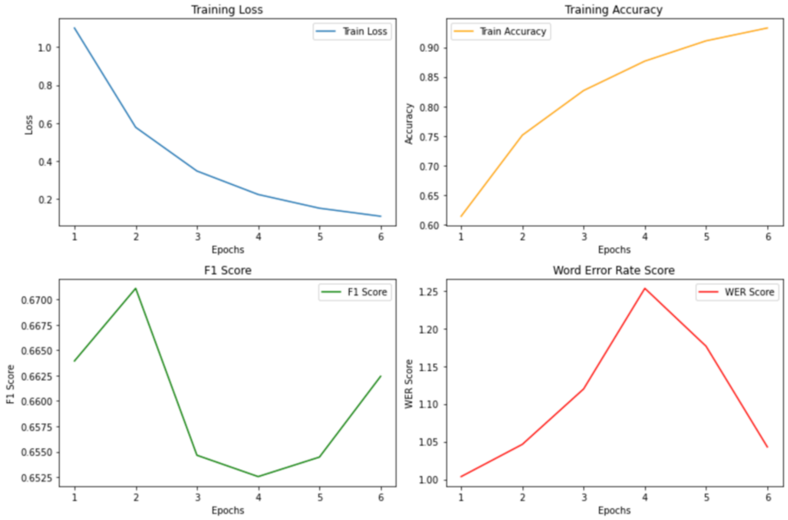


**Other Pretrained Model:**
I have checked another pretrained model i.e., **deepset/bert-base-cased-squad2.**

Train Accuracy (No Noise): 0.9326
Train Loss (No Noise): 0.1086

| | No noise | Noise V1 | Noise V2 |
|---|---|---|---|
| F1 Score | 0.6624 | 0.3788 | 0.2965 |
| Word Error Rate | 1.0429 | 2.3014 | 3.2682 |

Below is the plot of the other pretrained model (no noise) depicting Train Accuracy, Train Loss, F1 Score and WER over epochs.



**Observations:**

- The evaluation of the **distilbert-base-uncased** base model and its improvements on the SQAD dataset under different noise conditions reveals several observations.

- Implementing a **doc_stride of 128** led to slight improvements in F1 scores across all noise levels, indicating better handling of long documents.
- The introduction of **learning rate decay** with an exponential rate of 2e-2 further improved F1 scores, especially under noisy conditions, suggesting enhanced model fine-tuning.
- The most significant performance boost was observed with the **deepset/bert-base-cased-squad2** pretrained model, which achieved the highest F1 score and the lowest Word Error Rate (WER) in the no noise scenario.