CPSC 8430
Fall 2024
Homework 3
Extractive Question Answering
Abdullah Al Mamun

GitHub repository: **https://github.com/abdullm/CPSC_8430_Fall24_AM_HW3**
Trained BERT Model Location: **https://drive.google.com/drive/folders/1ZDoyP-_-o6o9kMtla5-JGkJLcpGa0o7b?usp=drive_link**

1. Introduction and Objective

The objective of this homework task is to develop and fine-tune a BERT model for extractive question answering on the SpokenSQuAD dataset. Extractive question answering involves identifying and extracting the correct answer span directly from a spoken or transcribed document. This task highlights the application of natural language processing (NLP) techniques to real-world noisy ASR data, testing the model's ability to handle audio-based data.

2. Model Architecture and Methodology

2.1. Dataset Overview

- SpokenSQuAD Dataset
- Contains transcribed spoken version of text-based SQuAD passages
- Training set: 37,111 question-answer pairs
- Testing set: 5,351 question-answer pairs
- Contains various levels of white noise

2.2. BERT Model Architecture for Question Answering

- *Pre-trained model used:* bert-base-uncased (https://huggingface.co/google-bert/bert-base-uncased)
- *Input Format:* Text tokenization with specific BERT tokens ([CLS], [SEP]), embedding of questions and passages
- *Tokenization:* Pre-processes the SpokenSQuAD dataset, splitting text into subwords and converting to IDs compatible with BERT's tokenization
- *Attention Mechanism:* The self-attention mechanism in BERT allows it to model complex relationships between words, which is used for determining answer spans within the passage

- *Training:* Fine-tunes BERT on SpokenSQuAD, predicting the start and end positions of answer spans
- *Output Layer:* The BERT model outputs two scores per token, indicating the likelihood of that token being the start or end of the answer span
- *Testing and Windowing Strategy:* Applies a sliding window strategy during testing, handling longer documents by splitting them into overlapping windows

2.3. Training Configuration

- *Learning Rate Decay:* A linear learning rate decay was applied, decrementing the learning rate across training steps to aid convergence
    - Initial Learning Rate: 2e-5
- *Batch Size and Gradient Accumulation:*
    - Batch Size per Step: 16
    - Gradient Accumulation Steps: 2
    - Effective Batch Size: 32
- *Doc Stride and Windowing Strategy:* To manage long passages, overlapping windows were created with a specified doc_stride, allowing the model to capture answer spans near the window edges
    - Max Sequence Length: 512
    - Doc Stride: 128

3. Evaluation and WER

3.1. Evaluation Procedure:

The model was evaluated on three variants of the SpokenSQuAD test set, each featuring different levels of background noise. This evaluation allows for an assessment of the model's robustness to transcription errors, measured through the Word Error Rate (WER). WER can be computed as:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

Where, S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference (N=S+D+C)

3.1.1. Test Datasets

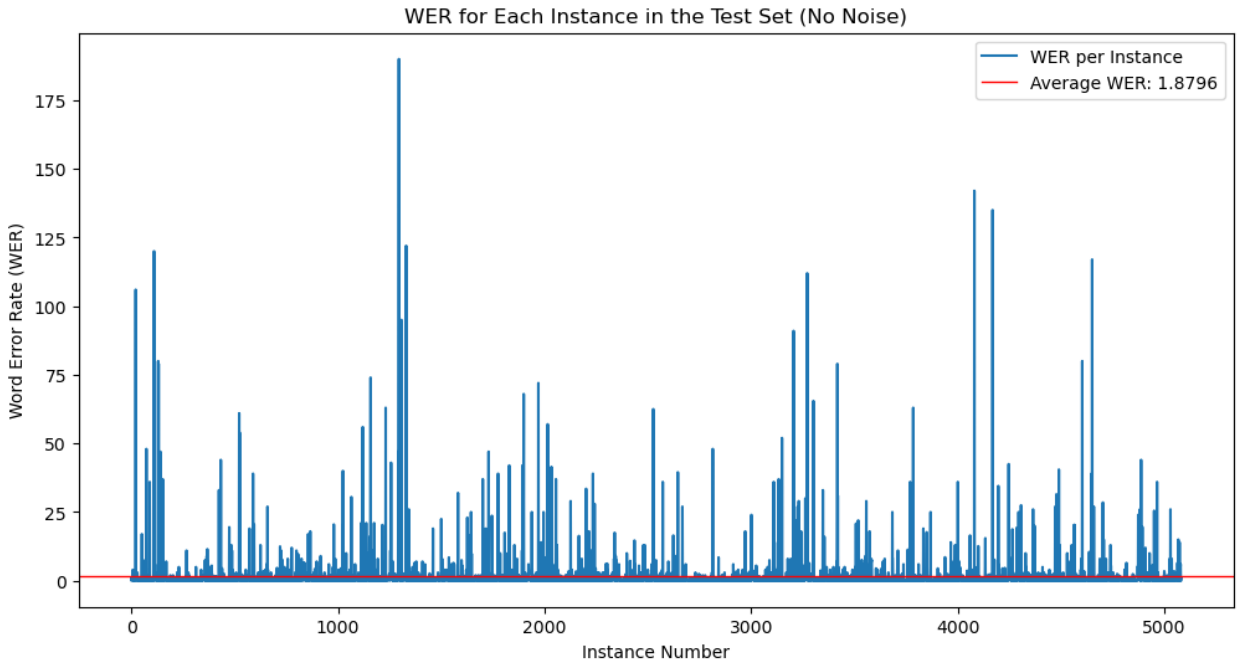
- WER23 (No Noise) Test Dataset
- WER44 (Noise V1) Test Dataset
- WER54 (Noise V2) Test Dataset

3.1.2. Evaluation Metrics

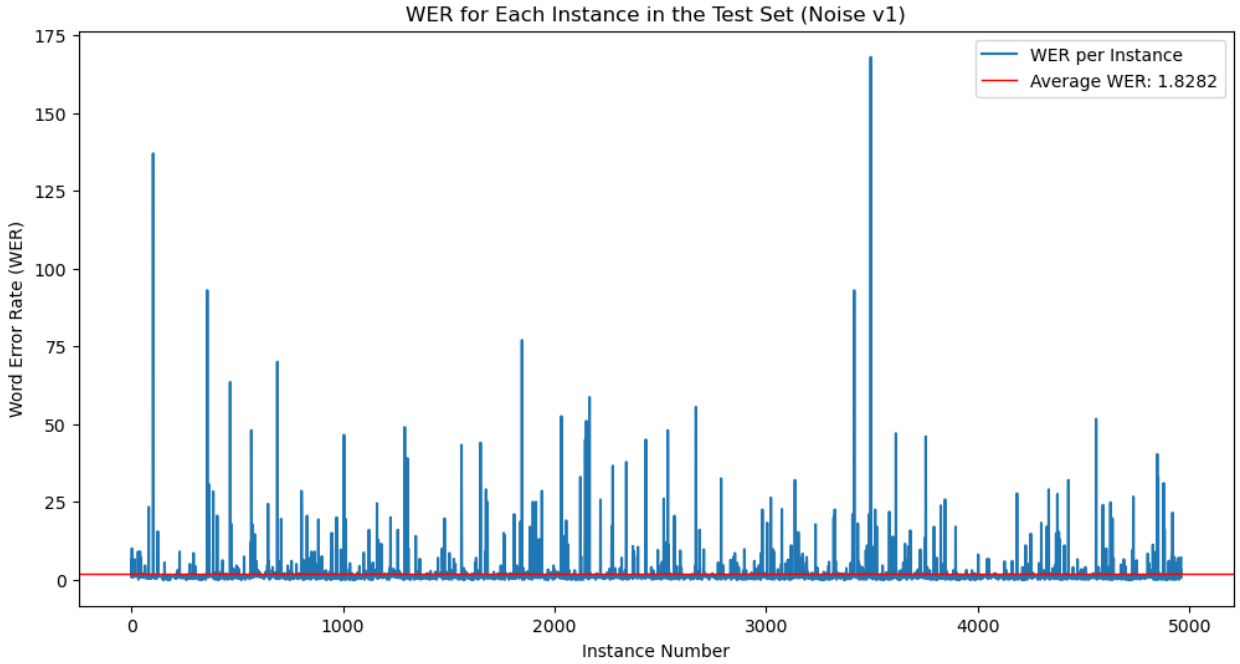
For each test instance, the WER was calculated between the predicted answer and the reference answer. An average WER across instances was also computed for both test datasets to summarize performance.

4. Results

4.1. WER23 (No Noise) Test Dataset



4.2. WER44 (Noise V1) Test Dataset



4.3. WER54 (Noise V2) Test Dataset