# CpSc 8430: Deep Learning
# Project 3

Giovanni Martino (giovanm@clemson.edu)

April 1, 2023

## 1 Introduction

The subject of this project is fine-tuning a BERT language model on the Spoken-SQuAD dataset [2], to perform the task of extractive question-answering on data containing transcription errors.

The code for this project can be found at the following github repository, under the hw3/ folder:
`https://github.com/rein5/cpsc8430-deep-learning`

The fine-tuned model was uploaded to huggingface, and can be found at:
`https://huggingface.co/rein5/bert-base-uncased-finetuned-spoken-squad`

The above github repository contains the Jupyter notebook *hw3/hw3.ipynb*, which will download the fine-tuned model from huggingface and evaluate it on the three Spoken-SQuAD test data splits. The notebook also contains code to keep training the model for additional epochs.

## 2 Model Architecture

The model was derived from the following pre-trained BERT [1] model:
`https://huggingface.co/bert-base-uncased`
This is the *base* variant of BERT, trained on uncased English input text.

As the final model component, a custom linear layer was added, mapping from the 768 BERT output features to 2 values, which are the start and end indices of the answer for a given example.

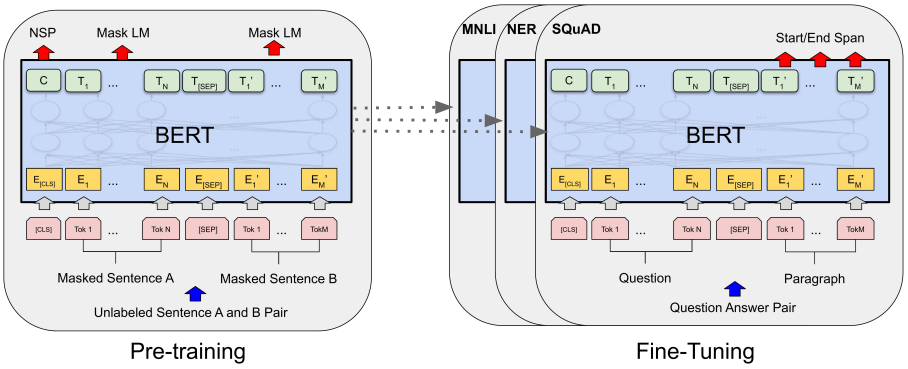


Figure 1: BERT fine-tuning [1]

On top of the *bert-base-uncased* pre-trained model, the corresponding tokenizer is also used to tokenize the Spoken SQuAD data, which is processed into examples of the form:

*Input: [CLS] question [SEP] context [SEP]*
*Output: (start index, end index)*

# 3 Data Processing

In addition to the aforementioned tokenization, the following techniques have also been used to pre-/post-process the data:

- Context Windowing: context sequences are split into multiple overlapping windows, with a stride of 64 tokens
- Truncation: input sequences are truncated to a max length of 384 tokens
- Padding: input sequences are also padded to the above max length, if shorter
- Out-of-context labels: if the answer is outside of the windowed context, the output label is set to (0, 0)

During post-processing, we skip predicted answers for which start index > end index, or answers whose length is greater than 32 (max_answer_length). For a given example ID, we pick the predicted answer with best score (defined as the sum of start and end logits).

# 4 Fine-Tuning

Starting from the pre-trained base model weights, the Question-Answering model was fine-tuned on the Spoken-SQuAD train split for 2 epochs.
The following tricks have also been used to improve training:

- Learning Rate Scheduling: learning rate gradually decreased from the initial value of $2 * 10^{-5}$, down to 0. This was implemented with a linear scheduler from the huggingface transformers library
- Mixed Precision: fp16 training, implemented using the huggingface accelerate library
- Multi-GPU training: this is also available via the accelerator object
- Optimizer: AdamW
- Batch Size: 8

# 5 Results

The fine-tuned model was evaluated on the three Spoken-SQuAD test splits, with the following results:

| **Data** | **F1 Score** |
|---|---|
| Test (no noise, 22.73% WER) | 74.29 |
| Test (V1 noise, 44.22% WER) | 54.94 |
| Test (V2 noise, 54.82% WER) | 41.41 |

# References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Chia-Hsuan Lee, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee. Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension. *Proc. Interspeech 2018*, pages 3459–3463, 2018.