In MCTest, we tried to use pretrained Bert to be our model. There are three type of our source data in MCTest such as training data, testing data, dev data. Also, there are given answer files for the training data and dev data.

First of all, we used pandas to read the source files that are the tsv type files. We convert them into a dataframe format and merge the training data with the training data answers. Since this is a MCTest task, the answers are represented by “A”, “B”, “C”, “D”. Therefore, we build a linkage with the answers(labels) and the details of the answer that means the meaningful sentence such as “go to school by bus”.

It is because the dataframe format dataset is not suitable to be the input format for fine-tuning the Bert. Therefore, we converted it into a list format and separate the context, question, choices(“A” “B” “C” “D”), labels and details of answers. Preparing for fine-tuning the Bert model. Also, we built a training dataset

After the Data preparation for fine-tuning, we started set up CPU and moved the model over to detected device. In order to reduce the probability of overfitting, we initialize the adam optimizer with weight decay. Also we initialize the data loader for the training data .

Then, we activated training mode of model and pull all the tensor batches required for training. Also, we did a encoding for working on the tokenizer for the training data. However, we encounter some difficulties such as we can’t get the google module and there is a label issue that make the MCTEST can’t calculate and extract the loss for every parameter that needs grad update. That affected the model evaluataion.