Overall, this is a model that is based off the traditional BERT, which excels at question answering. While BERT’s performance on normal question answering is approaching that of humans, the upgraded model we implemented is specifically tailored to natural questions, which are thought to be more challenging for language models than the regular question sets.

My implementation is in total four steps, very similar to the structure of our original BERT:

* Data loading and preprocessing. The data are readed in with the built-in load_dataset function and are stored as a self-defined dataset which I call QADataset.

* Model loading. Here I define a new version of Bert model stemming from the pretrained version distilbert-base-uncased.

* Train. This is where we customize how many epoch to run. Within each epoch, we first fetch batch data and find out the true labels (start_position, end_position, answer_type). Then we use the model to predict the labels, calculate loss in accordance with the paper, and lastly backward pass the loss to tune parameters.

* Eval. Here is where we evaluate how good the model performance is. The precision, recall and f1 score are accumulated throughout the validation batches and taken an average on by the end. In each batch, we get the predicted value and compare with the actual value. It is a little tricky to calculate how much the two lists overlap and not overlap, so I used a python hack to calculate the joint and difference.

To run the model, simply start the block containing the main function. That’s the one and only entry point for this model. The main function finishes the whole cycle of all the steps mentioned above. Below the main function I also wrote a log function so that I can get a snippet on the result quality without the need to finish all training.

There are no known bugs from the code, but I did notice some predicted start is greater than the predicted end.

In the end, the model achieved a precision of 0.67, recall rate of 0.70, and an overall f1-score of 11.32(And now I realized my f1 score is off). It is achieved with a batch_size of 32, learning rate 5e-5 and there are 2 epochs in total. It is noteworthy that this result should be higher than the actual accuracy as we are using the validation set as the evaluation set. It is very likely that the model has overfitted to the dataset and will perform worse than it should be on new datasets.

Video:https://drive.google.com/file/d/1EYU94GJDIPpjVfNFbuV6le809yR_fZSZ/view?usp=sharing