Report Question 1

In this question. I used Bart and T5 for Generative Question Answering task. Both architecture is encoder-decoder model.

Detail:

Because the size of dataset is large(272k). So I use only 30k examples for training both model. I have tried using 50k, 80k and all example. But the Ram in Kaggle was overflowed and the session ended.

(I think my code is not optimized because during traning. The usage of cpu taken the most of time. I think the reason is because the load example function before feeding to model take much cpu usage.

Therefore, I tried to tokenized the whole dataset. But for some reason, it took a much longer than previous methods.

Also, so I followed the method tokenized the batch before feeding in since many tutorial I have seen use this way.)

Traning process: I use GPU t4x2 on Kaggle for both model. The maximum epoch is 3. The number of document I used is 3(select the first 3 docs in each question). Because one question can have many answers. So I narrowed down to only maximum 3 answers for each question.

I coded in pytorch with hugging face library. Used the model architecture in huggingface.

* BartForConditionalGeneration pretrained(facebook/bart-base) and T5ForConditionalGeneration pretrained(t5-base)

I used pytorch lightning for training. Pytoch traning has many advantages to traditional training, it has many strategies that support during training such as schedulers.

Tokenizer: I used the batch size of 16, the max\_length for context, question is 256 tokens, truncation enable. For answer the max\_length is 16 tokens

Optimizer: AdamW with learning rate 2e-4

The training process took about 12 hours on Kaggle. Unfortunately, my model only save the first checkpoint at iteration 4119 at 7th hour because I set the Trainer to save at the end of each epoch.

I have post the training process in tensorboard, you can check them at view\_tensorboard.py.

https://tensorboard.dev/experiment/PsK6CvQYSeKbv4F6L3jQbQ/?fbclid=IwAR3RiTX8HiDJoZfYttqObZ\_SbQ7\_BeX7mDDUwKTVF0pNF3Gb-4NtTm9OvIY#scalars

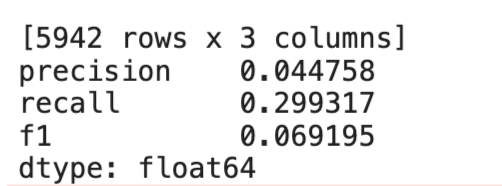
I think the model is underfitting because for most of time, it stuck around 3 -> 3.5 and also the training example is limited.

About the code:

Untill this day, I have done t5 model code. The detailed is:

Evaluation.py :load the model and tokenizer and begin compute the rougeL score. The result is precision, recall and f1 score of rougeL score.

Result on t5-base on validation set:



Generate.py: this file is use for user example. You can change the content of question and context to see the result. (because the answer only have 16 token at most during training. So I think if we set the max\_length >16, the model will produce trivial answer).

The result of this above example is taken from a short text inside the context (The different wave length of light correspond to different color…)

(\*until now I have tried to use 125 tokens for answer but the loss is worser than 16 tokens?)

The train\_with\_lightning.py is the script for fine-tunning the model. There is two option 1 is train with pytorch-lightning library and the second is train with traditional method. (loss.backward(), optimizer.step()…..). The code for second method is absolutely longer than the first one.

Utils.py: is the file save all the functions and class to support other files.

The reason that I did not do remain work with bart because it loss is no better than T5. Now I am trying to improve the rougeL socre by making my model train all the examples(it crashed all the time due to limit RAM used) and read other paper to find other architecture to solve this problems such as multi-task seq2seq and Route Transformers.

For question 2: I have seen that t5 is a multi-task text-text model. So models in huggingface used t5 to question generation.

(To be honest, I am not reading paper most time, I used time to reading tutorial and figure how to code with library)