Homework #3 Report

Applied Deep Learning

資工碩一　張凱庭　R10922178

Q1: Model

1. Describe the model architecture and how it works on text summarization

Model Configuration


```
{
  "_name_or_path": "google/mt5-small",
  "architectures": [
    "MT5ForConditionalGeneration"
  ],
  "d_ff": 1024,
  "d_kv": 64,
  "d_model": 512,
  "decoder_start_token_id": 0,
  "dropout_rate": 0.1,
  "eos_token_id": 1,
  "feed_forward_proj": "gated-gelu",
  "initializer_factor": 1.0,
  "is_encoder_decoder": true,
  "layer_norm_epsilon": 1e-06,
  "model_type": "mt5",
  "num_decoder_layers": 8,
  "num_heads": 6,
  "num_layers": 8,
  "pad_token_id": 0,
  "relative_attention_max_distance": 128,
  "relative_attention_num_buckets": 32,
  "tie_word_embeddings": false,
  "tokenizer_class": "T5Tokenizer",
  "torch_dtype": "float32",
  "transformers_version": "4.19.0.dev0",
  "use_cache": true,
  "vocab_size": 250100
}
```


The “mt5-small” is an encoder-decoder pre-trained multi language model. For text summarization, the context(article) is fed into the encoder part.Then the encoded hidden states via cross-attention layers to the decoder

and auto-regressively generates the decoder output and stop until a <EOS> token appears.

2. **Preprocessing**

   Using the mt5 tokenizer which was based on SentencePiece. to tokenize the news and title. If the length is exceeded, then it will be truncated.

Truncation	
max input length	256
max target(output) length	64

Q2: Training

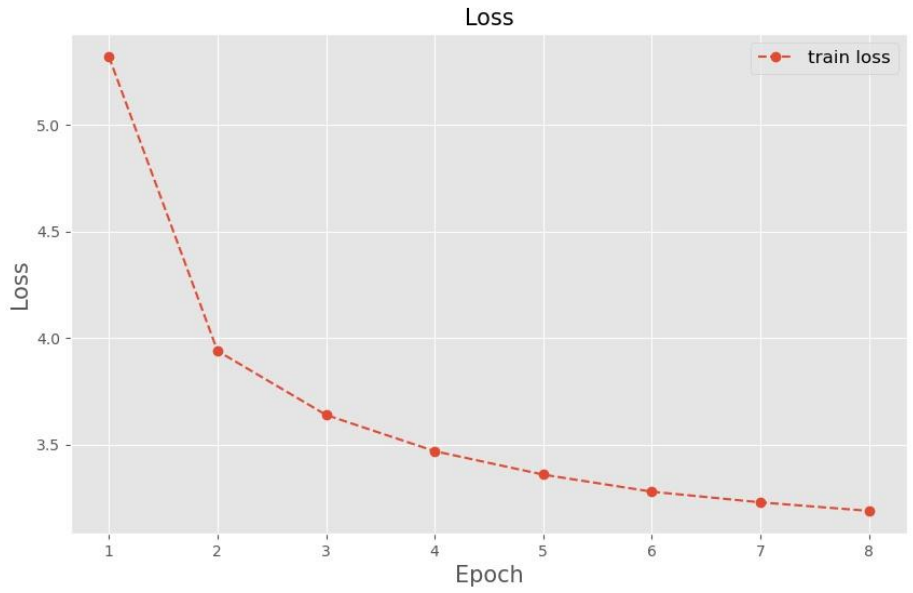
1. **Hyperparameter :** Describe your hyperparameter you use and how you decide it.

Hyperparameter	
pretrained model	google/mt5-small
loss functoin	CrossEntropyLoss
optimizer	AdamW
learning rate	5e-5
batch size	4
num epochs	8

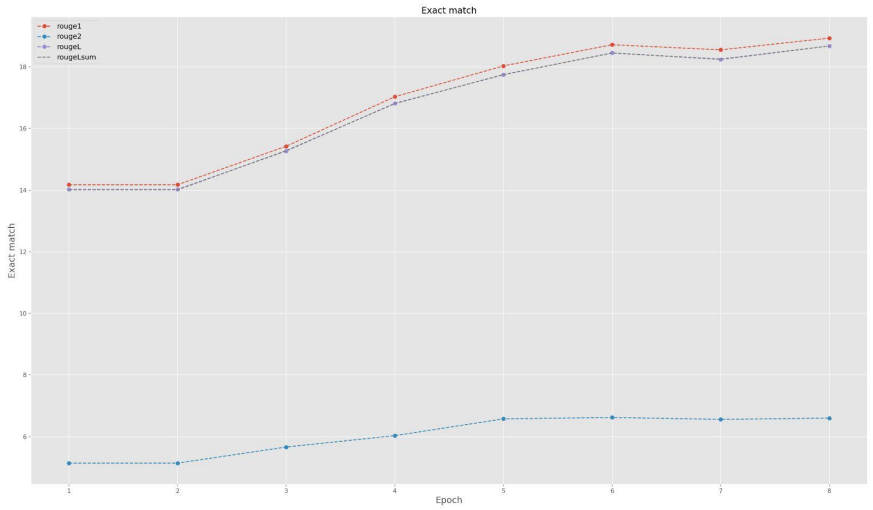
   Using the hyperparameter recommended from: huggingface/transformers, but increase the number of epochs.

2. **Learning Curves**

Loss curve



Rouge curve



(rougeL and roougeLsum are overlapped)

Q4: Generation Strategies

1. **Strategies:** Describe the detail of the following generation strategies
    a. Greedy: When generating, always select the highest probability as the next word.
    b. Beam Search: Keeps track of the second(or the number of beams) most likely one.

c. Top-K sampling: K most likely next words are filtered and the probability mass is redistributed among only those K next words.
d. Top-P sampling: sampling chooses from the smallest possible set of words whose cumulative probability exceeds the probability P.
e. Temperature: a trick to make the probability sharper (increasing the likelihood of high probability words and decreasing the likelihood of low probability words), similar to softmax.

2. **Hyperparameters**:

Setting	rouge-1	rouge-2	rouge-L
greedy	24.82	9.437	22.26
beams=2	25.78	10.25	23.12
beams=4	26.12	10.62	23.47
temperature=0.7	21.58	7.665	19.30
temperature=0.9	17.66	5.823	15.88
top 50	19.60	6.422	17.37
top 30	20.55	6.893	18.25

Final generation strategy: beams = 4