ADL 2024 HW1 Report

陳彥廷，資管四，b10705013

Q1: Model

Q1-1 : Describe the model architecture and how it works on text summarization

T5 (Text-to-Text Transfer Transformer) is proposed by Google Research in 2020, which is an encoder-decoder model based on the Transformer architecture and trained by large datasets.

In pretrained task, T5 compares many methods in high-level approaches、corruption strategies、corruption rate and corrupted span length. Finally, it decides to use the BERT-style masked language model (MLM)、replace span、25% corruption rate and 3 tokens corrupted span length.

In text summarization task, T5 will first add a prefix "summarize: " to the input text, then the encoder will convert the input text into a sequence of hidden states, and the decoder will generate the output text based on the hidden states until the end marker or the maximum length is reached.

In this project, I use the `google/mt5-small` model, which is a smaller version of T5, to generate multi-lingual text summarization.

Q1-2 : Describe your preprocessing

First, I add a prefix "summarize: " to the input text. Then, I use the default tokenizer of the model to tokenize the input text and the output text.

The mT5 tokenizer is based on the SentencePiece model, a subword tokenization technique that can handle text across multiple languages. It does not rely on language-specific rules, making it ideal for multilingual tasks.

When we input a text to the tokenizer, it will first split the text into sentences, then tokenize each sentence into subwords, and finally convert the subwords into token IDs.

Q2: Training

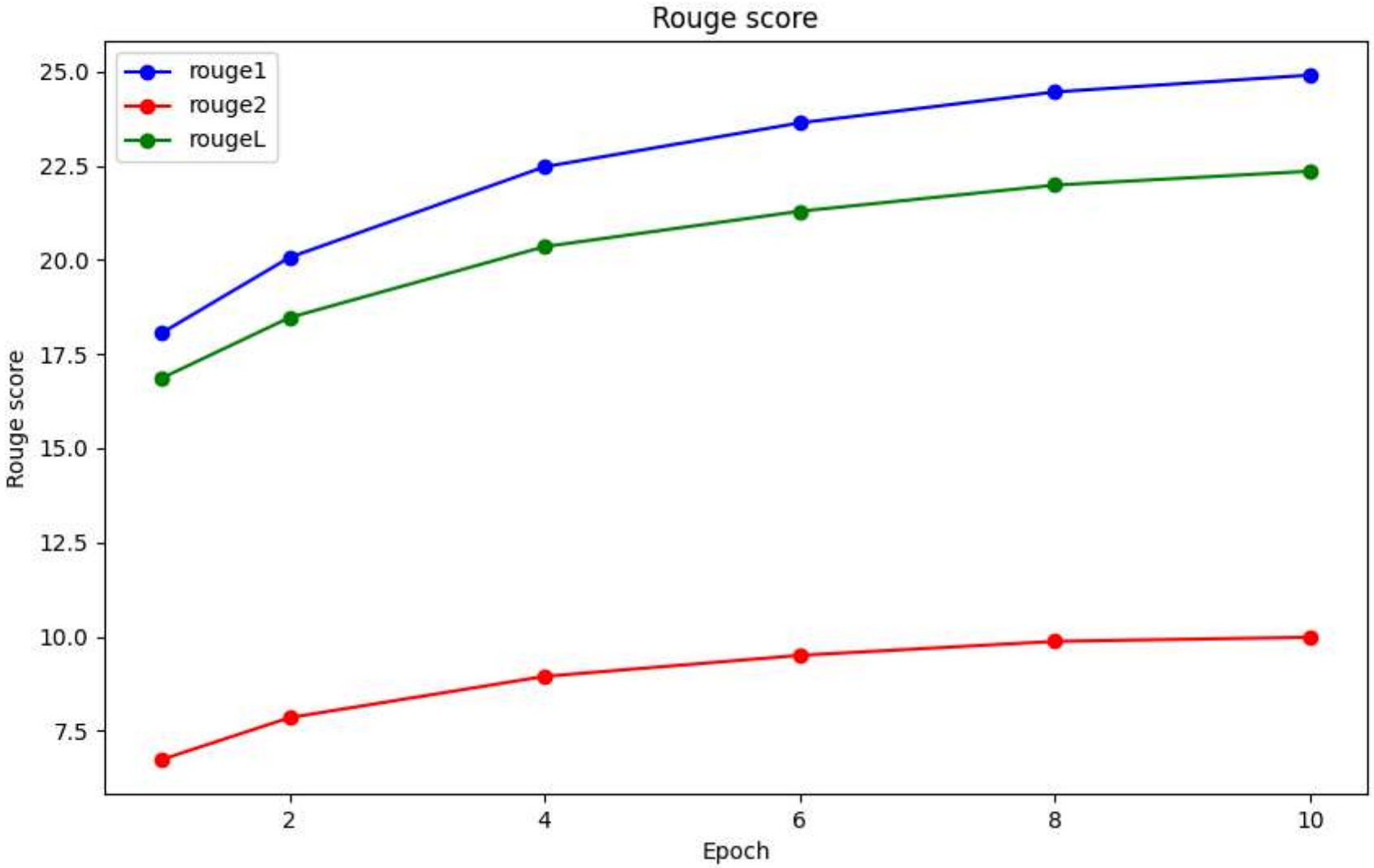
Q2-1 : Describe your hyperparameter you use and how you decide it

Hyper Parameter	google/mt5-small
Optimization Algorithm	AdamW
lr_scheduler_type	cosine
max_source_length	512
max_target_length	64
num_bean	20
Learning Rate	1e-4
Batch Train Size per Device	4
Batch Eval Size per Device	8
Gradient Accumulation Steps	2
Epochs	10

- lr_scheduler_type : According to the test, cosine learning rate scheduler is better than linear learning rate scheduler.
- max_source_length & max_target_length : The max_source_length is set to 512 to support longer input text, and the max_target_length is enough for the output text because there are only 10 titles larger than 64 tokens in the training dataset.

- num_beams : The number of beams is set to 10 to generate more diverse outputs.
- Learning Rate : The learning rate is set to 1e-4, which is a common value for fine-tuning.
- batch size : According to the test, the smaller batch size is better than the larger batch size.

Q2-2 : Plot the learning curves (ROUGE versus training steps)



Q3: Generation Strategies

Q3-1 : Describe the detail of the following generation strategies

- Greedy : Always choose the token with the highest probability as the next token.
- Beam Search : Keep track of the top k sequences and extend each sequence with all possible tokens. Then, select the top k sequences from all possible sequences.
- Top-k Sampling : Randomly sample from the top k tokens with the highest probabilities.
- Top-p Sampling : Randomly sample from the smallest set of tokens whose cumulative probability exceeds the probability p.
- Temperature : Divide the logits by the temperature and then normalize them to get the probabilities.

Q3-2 : Try at least 2 settings of each strategies and compare the result

Strategy	Rouge1	Rouge2	RougeL
Beam 20	24.90	9.983	22.35
Top-p 0.9	19.66	6.360	17.42
Temperature 0.8	20.38	6.813	18.11
Temperature 1.2	16.11	4.454	14.19
Greedy	23.67	8.653	21.35

Q3-3 : What is your final generation strategy?

Beam search with 20 beams is the best strategy for text summarization because it can generate more diverse outputs and achieve higher ROUGE scores.

Reference

- ChatGPT
- Copilot