ADL HW3 Report

Q1:Model

A. Model(mT5):

- Architecture: The mT5 is a multilingual variant of Google's T5 model that was pre-trained over a dataset of more than 101 languages. The architecture of mT5 is close to T5. T5 is a Transformer based architecture that uses a text-to-text approach. Some of its pre-train tasks are similar to BERT but replacing the fill-in-the-blank cloze task with a mix of alternative pre-training tasks.
- how it works on text summarization: The input sequence is fed to the model using input_ids. The target sequence is shifted to the right, i.e., prepended by a start-sequence token and fed to the decoder using the decoder_inputs_ids. For sequence-to-sequence generation, it takes care of feeding the encoded input via cross-attention layers to the decoder and auto-regressively generates the decoder output.

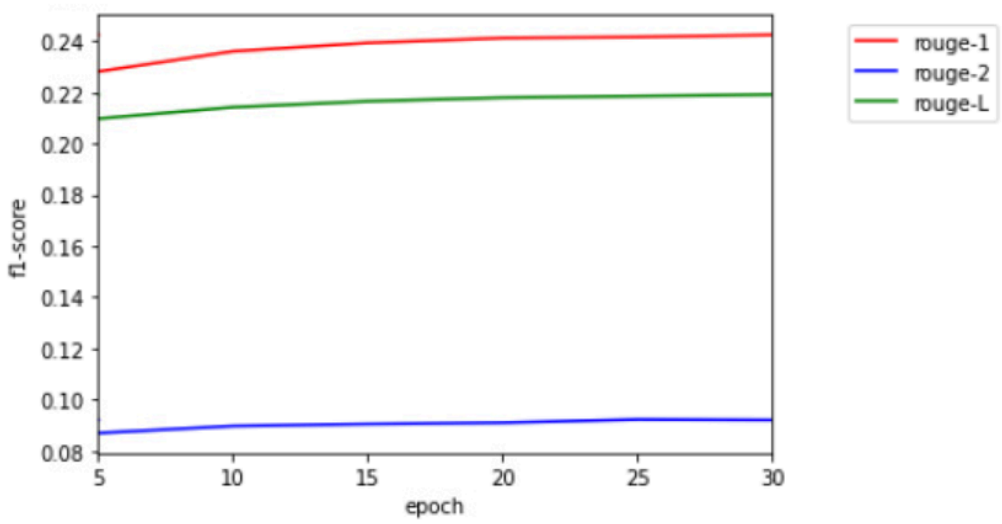
B. Preprocessing: Use pre-trained tokenizer from "google/mt5/small" to tokenize the "maintext" of data. For the labels("title" of data), use "as_target_tokenizer" of tokenizer to tokenize the labels.

Q2:Training

A. Hyperparameter:

- Tokenizer: The max_length of tokenizer of maintext is 712, and the max_length of target tokenizer is 120.
- Training: The batch size of training is 32. The optimizer is Adafactor. The initial learning rate is 3e-4 but the learning rate would $*$ 0.5 every 4 epochs. The number of epoch is 30.
- Generate: The max_length is 120, number of beam is 3, top_k is 30, repetition_penalty is 2.5, and length_penalty is 1.

B. Learning Curves: Epoch=30, calculate rouge score every 5 epoch.



Q3: Generation Strategies

A. Strategies:

a. Greedy: Greedy strategy simply choose the word with the highest probability as next word.

b. Beam Search: We need to set the beam size(k) when applying beam search. It reduces the risk of missing hidden high probability word sequences by keeping k most probable sequences at each time step and eventually choosing the sequence that has the highest probability.

c. Top-k Sampling: In top-k sampling, it filters the top k probable words and sample one of these k words by distribution.

d. Top-p Sampling: In top-p sampling, it picks minimum number of words that the cumulative probability exceeds the probability p and sample one of these words according to their distribution.

e. Temperature: Temperature is a hyperparameter to the softmax. By lowering the temperature, the probability distribution would be spiky (increasing the likelihood of high probability words and decreasing the likelihood of low probability words), otherwise, the probability distribution would be uniform.

B. Hyperparameters:

- Try 2 settings of each strategies and compare the results.

strategies	rouge-1	rouge-2	rouge-L	time
Greedy	0.2390	0.0809	0.2079	20 min
No greedy	0.1776	0.0463	0.1535	25 min
Beam Search (size=3)	0.2464	0.0967	0.2234	30 min
Beam Search (size=5)	0.2451	0.0977	0.223	38 min
Top-k Sampling (b=3, k=15)	0.2464	0.0967	0.2234	30 min
Top-k Sampling (b=3, k=30)	0.2464	0.0967	0.2234	30 min
Top-p Sampling (b=3, k=15, p=0.95)	0.2424	0.0930	0.2196	50 min
Top-p Sampling (b=3, k=15, p=0.8)	0.2426	0.0923	0.2200	50 min
Temperature (b=3, k=15, t=0.9)	0.2344	0.0892	0.2146	32 min
Temperature (b=3, k=15, t=0.95)	0.2404	0.0930	0.2192	32 min

- Final generation strategy: I combine beam search with beam size 3, top-k sampling with k=30.