ADL HW2 Report

姓名：陳竹欣

學號：B10902005

系級：資工三

Q1: Model

Model

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

我用的 model 是 `T5-small` 。

`T5` 使用的 model architecture 為 encoder-decoder 架構。Encoder 可以看到前一層全部的資料；Decoder 僅能看到前一層自己神經元位置前的資料。

在 text summarization 的任務中，輸入為完整的文章，也就是 maintext 的部份。maintext 會先作為 encoder 的輸入，產生 encoded sequence 之後，再作為 decoder 的輸入，最終預測 summary 。

Preprocessing

Describe your preprocessing (e.g. tokenization, data cleaning and etc.)

Step 1. Add prefix

要在 text 的最前面加上 "summarization: "，T5 才會知道現在要完成的任務是 summarization 而不是其他任務。

Step 2. Truncation

當輸入的長度大於指定的 max length 時，要把超過的字串切掉；反之要補 0 。

Step 3. Tokenization

把中文字元和英文單字轉成 token。

Q2: Training

Hyperparameter

Describe your hyperparameter you use and how you decide it.

arguments	value
`--max_source_length`	`1024`
`--max_target_length`	`64`
`--per_device_train_batch_size`	`4`
`--learning_rate`	`1e-4`
`--num_train_epochs`	`10`
`--gradient_accumulation_steps`	`4`

1. max source length

   我有嘗試兩種可能，分別為`256`以及`1024`，以下為使用 beam search (`num beam = 8`) 預測的結果：

max source length	rouge-1	rouge-2	rouge-l
256	24.067	9.262	21.146
1024	25.576	10.097	22.317

   雖然`256`就能過 simple base line，但在系統效能允許的情況下，盡量不做 truncation 肯定是更好的選擇。

2. max target length

   因為投影片裡面建議使用`64`，我在訓練的時候也沒有調整過，所以就是`64`了。

3. batch size

   我有嘗試兩種可能，分別為`4`以及`8`，以下為使用 beam search (`num beam = 5`) 預測的結果：

batch size	rouge-1	rouge-2	rouge-l
4	21.747	7.839	19.167
8	18.758	6.373	16.665

   這是我比較前期的嘗試，所以忘記當時的超參數是多少了，可能是因為當時的`epoch=5`，導致效果不太好，但顯然把 batch size 調大對於最終結果並沒有幫助，反而還讓分數變低，所以在之後的測試都是用 batch size = 4 進行訓練。

4. learning rate

   我有嘗試過 `1e-3`, `1e-4`, `5e-4` 以及 `1e-5`，除了 `1e-3` 有點太大導致效果不好之外，其他的似乎都差不多，所以最終選擇 `1e-4`。

5. epoch

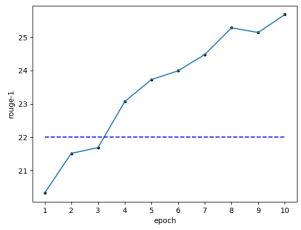
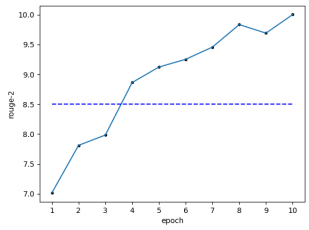
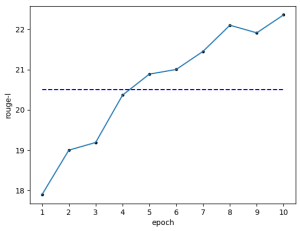
   我有嘗試過 `5` 以及 `10`，以下為使用 beam search (`num beam = 8`) 預測的結果：

epoch	rouge-1	rouge-2	rouge-l
5	22.056	8.274	19.512
10	24.066	9.262	21.145

Learning Curves

Plot the learning curves (ROUGE versus training steps)

以下皆為使用 beam search (`beam size = 5`) 預測的結果

rouge-1	rouge-2	rouge-l
		

詳細的數據以及作圖的程式碼在 `/b10902005/Q2-learning_curve/result_e{i}`。

Q3: Generation Strategies

Stratgies

Describe the detail of the following generation strategies:

Greedy

在預測下一個字的時候，直接選機率最大的。

問題：局部的最佳解不一定是整體的最佳解。

Beam Search

假設 $\text{beam size} = k$，在預測下一個字的時候，會考慮機率前k高的可能。也就是當 $k = 1$ 時，和 greedy 是一樣的。

問題：當 k 太大時會跑太久，而且會選到太 general 的答案，k 太小又會和 greedy 遇到一樣的問題。

Top-k Sampling

在預測下一個字的時候，會把機率前k高的可能做 sampling 。也就是當 $k = 1$ 時，和 greedy 是一樣的。

問題：當 k 太大時可能會 sampling 到一些奇怪的字（也就是機率比較低的字），導致語意偏離，k 太小又會和 greedy 遇到一樣的問題。

Top-p Sampling

在預測下一個字的時候，會把機率出現在前 p 的可能做 sampling 。

優點：由於在 narrow distribution 的狀況，我會希望 k 小一點，而在 board distribution 的狀況，我會希望 k 大一點。所以透過選取機率出現在前 p 的可能，我可以做到動態的調整 k 進而得到更通順的語句。

問題：當 p 太大時會跑太久，也可能會 sampling 到一些奇怪的字，p 太小又會和 greedy 遇到一樣的問題。

Temperature

在計算 softmax 的時候，會把向量除以一個值 τ 。

$\tau \uparrow$ 代表 $P(w)$ 會變得更加 uniform，下一個預測的字的可能也會更多元 。

$\tau \downarrow$ 代表 $P(w)$ 會變得更加 spiky，下一個預測的字的可能也會更保守 。

Hyperparameters

Try at least 2 settings of each strategies and compare the result. What is your final generation strategy? (you can combine any of them)

以下的測試中，generation controlling 的參數如下：

arguments	value
`temperature`	`0.5`
`max_length`	`64`
`min_length`	`15`
`repetition_penalty`	`10`
`length_penalty`	`1.0`
`no_repeat_ngram_size`	`2`

beam search

num beam	rouge-1	rouge-2	rouge-l
5	25.682	10.004	22.356
8	25.576	10.097	22.317

其實兩者沒有差很多，但因為預測的時間限制為一小時，`num beam = 8`的執行時間在邊界，很有可能不小心就超時了，所以保險起見我選擇`num beam = 5`即可。

top-k

k	rouge-1	rouge-2	rouge-l
10	24.064	8.121	20.377
20	23.897	8.422	20.208

top-p

p	rouge-1	rouge-2	rouge-l
0.25	23.992	8.065	20.309
0.5	23.978	8.086	20.340
0.75	23.540	7.886	20.007

top-k + top-p

k	p	rouge-1	rouge-2	rouge-l
50	75	23.431	7.770	19.864

我最終選擇了 beam search (`num beam = 5`) 作為最終的 generation strategy 。

詳細的數據以及測試的程式碼在`/b10902005/Q3-Testing`。