M11152014　蘇珉淙

Q1: Model (2%)

- ❖ Model (1%)
  - ➢ Describe the model architecture and how it works on text summarization.

這次作業是 fine-tune mt5-small model，使用的 pre-trained model 是 huggingface 上的 google/mt5-small，架構為 transformer 的 encoder-decoder 類型，將新聞文章做 tokenization 以及 padding 後送進 encoder，接著將 encoder 的輸出包含 <BOS> token 一同輸入到 decoder 做 conditional language modeling，過程中持續地去 predict 下一個字出現的機率，而最後 decoder 輸出的 token sequence 即是 text summarization 的結果。

- ❖ Preprocessing (1%)
  - ➢ Describe your preprocessing (e.g. tokenization, data cleaning and etc.)

將新聞文章使用 T5Tokenizer 進行 tokenization，先對 maintext 做 tokenization，設定 padding 為 max_length = 1024，再對 title 做 tokenization 時，則設定 max_length = 128。

Q2: Training (2%)

❖ Hyperparameter (1%)

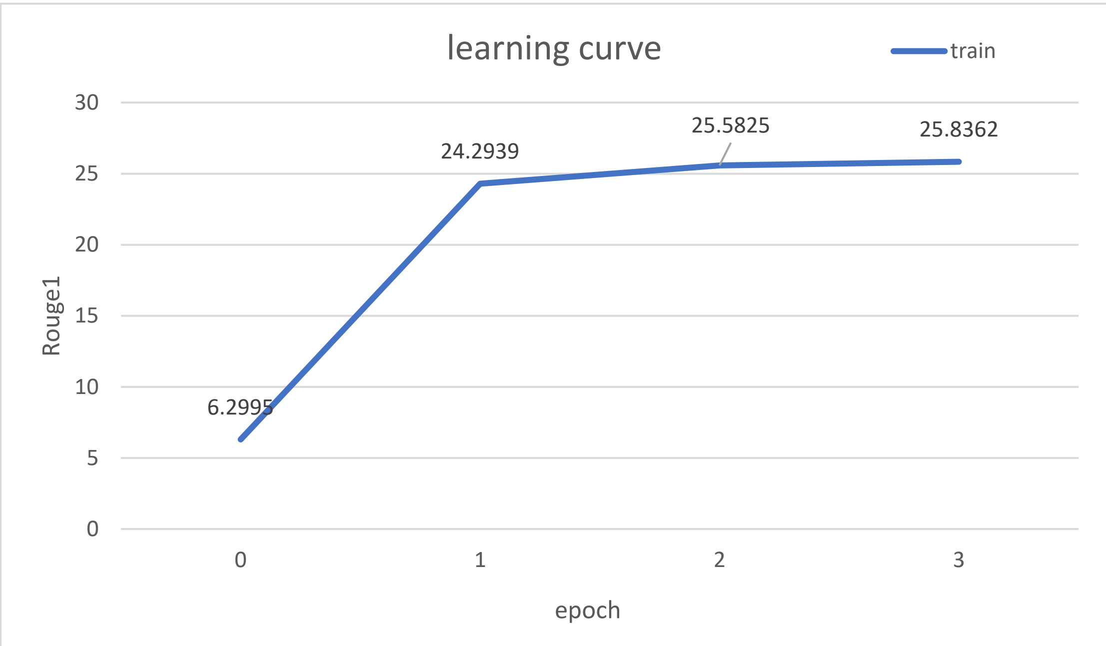
➢ Describe your hyperparameter you use and how you decide it.

我所使用的 hyperparameter 為 batch size = 4、learning rate = 1e-4、warmup ratio 保持預設 0.1，總共訓練了 3 個 epoch，tw_rouge 得分結果如 Q3 第二題中的表格所示，紅色標示為最佳結果。

❖ Learning Curves (1%)

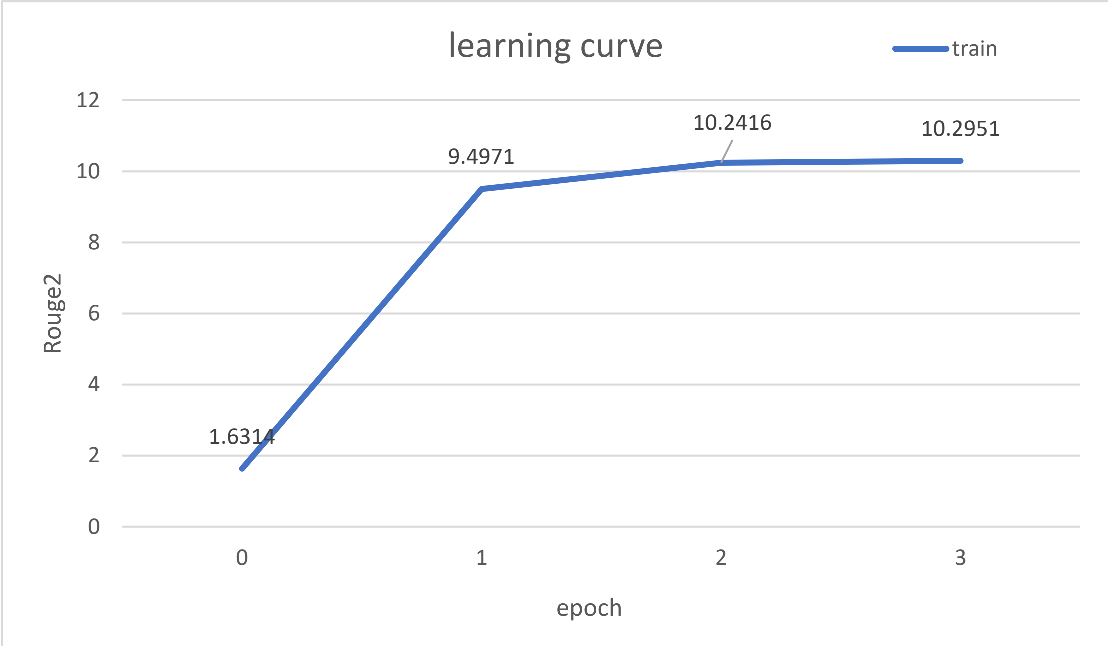
➢ Plot the learning curves (ROUGE versus training steps)

Rouge-1

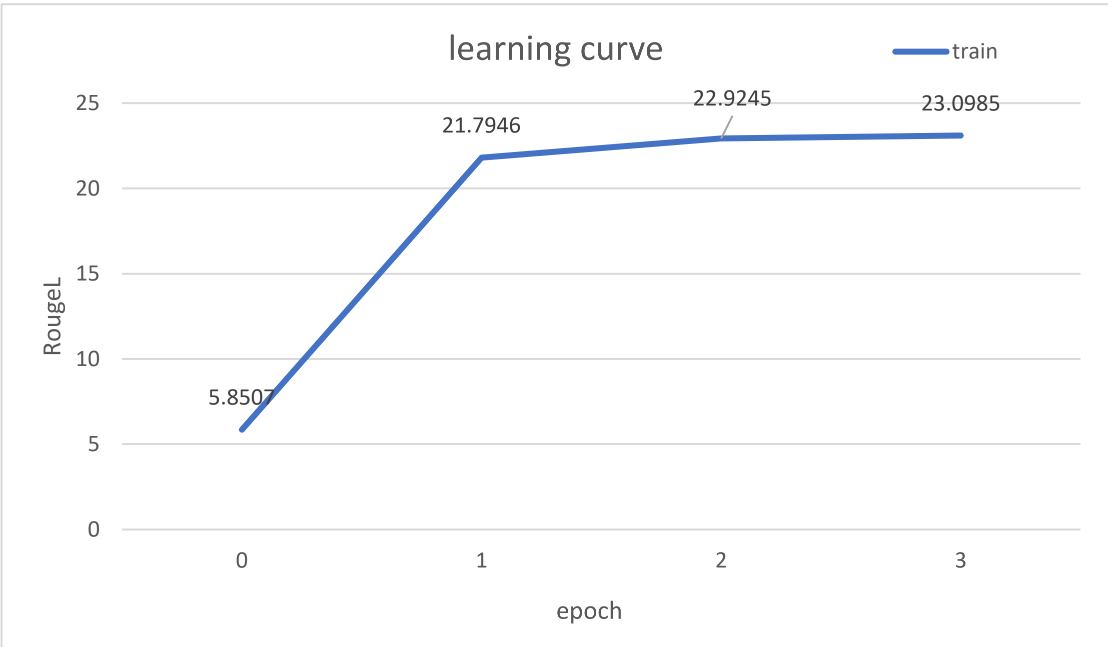


M11152014　蘇珉淙

Rouge-2



Rouge-L



Q3: Generation Strategies(6%)

- Stratgies (2%)
  - Describe the detail of the following generation strategies:
    - Greedy: 在每個時間點，都只選擇機率最高的 token。
    - Beam Search: 追踪前 k 個最有可能的 sequence，並選擇較好的一個，但這種方法可能需要龐大的計算量。假設用 n 個變數來生成一個 token length 為 m 的句子，它將會是 n^m 之多。
    - Top-k Sampling: 此方法是用來確保可能性較小的詞被 sample 到的機率很低，並確保每次 generation 只考慮前 k 個可能的 token。
    - Top-p Sampling: 類似 Top-k Sampling，但它只關注機率總和 $\geq p$ 的最小詞組。
    - Temperature: Sampling with temperature 會隨機選擇字詞，並透過設定 temperature ($0 < temp \leq 1$) 來強調機率高的 token，同時減少機率低的 token。
- Hyperparameters (4%)
  - Try at least 2 settings of each strategies and compare the result.

測試 Beam Search 並且不使用 sampling

Beam Size	Rouge-1	Rouge-2	Rouge-L
1 (Greedy)	24.3216	8.8955	21.774
4	27.1313	10.8587	24.2078

測試 top-k sampling (beam size:1; top-p: 1.0 ; temperature: 1.0)

top-k	Rouge-1	Rouge-2	Rouge-L
50	20.1035	6.1276	17.5443
80	19.5703	6.0945	17.2052

測試 top-p sampling (beam size:1; top-k: 50 ; temperature: 1.0)

top-p	Rouge-1	Rouge-2	Rouge-L
1.0	20.1035	6.1276	17.5443
0.5	24.8409	9.0044	22.0125

測試 sampling with temperature (beam size:1; top-k: 50 ; top-p: 1.0)

temperature	Rouge-1	Rouge-2	Rouge-L
1.0	20.1035	6.1276	17.5443
0.1	25.7875	9.6935	23.0821

- What is your final generation strategy? (you can combine any of them)
  考量 performance 與 runtime，我最後選擇使用 Beam Search，beam size 為 4 且不使用 sampling，當我的 generation strategy。

Bonus: Applied RL on Summarization (2%)

- Algorithm (1%)
  - Describe your RL algorithms, reward function, and hyperparameters.
- Compare to Supervised Learning (1%)
  - Observe the loss, ROUGE score and output texts, what differences can you find?