1.Model

本次2b作業中，我是使用t5-small-chinese-cluecorpussmall 作為T5的model

而gpt2的model是使用gpt2-distil-chinese-cluecorpussmall

以上兩者都是用CLUECorpusSmall 來pre-trained的中文model

其中T5是由encoder與decoder組成，擅長seq to seq任務，像是問答、翻譯、摘要…，混合unsupervised和supervised進行pre-trained

而GPT2 只有decoder，擅長文字生成，並依據前向文本的context生成文本，使用unsupervised 來pre-trained。

2.Dataset

我在T5model使用了getitem來處理input data，他將input設為text, 而label 設為summary，並且對他們encode再轉為 PyTorch tensor，之後對output的 token id list做 padding，使結果有相同長度。其中input-padding 的token id設為tokenizer.pad\_token\_id，label-padding 的token id設為-100從而使模型不計算padding的loss

而我在GPT2model使用了三個function處理input data ，getitem、getitem2、collator\_fn

首先，在Getitem中，他將 input設為(文本token id list) +(SEP)+(摘要token id list)

再將label設為 -100 \* len(文本token id list+) +(摘要token id list )

我想利用這個順序使模型學到SEP 代表摘要開始，因此就能在evaluate時輸入文本token id list + SEP  後，讓模型知道要產生摘要。而label設為 -100 \* len(文本token list+SEP) +(摘要token id list)，是為了用-100讓模型只計算摘要部分的loss

再來，getitem 2，他將input設為(文本token id list) +(SEP)，

在將label設為 -100 \* len(+文本token id list+SEP) +(摘要token id list)

最後，collator\_fn，將 input和 label 中的每個元素轉換為 PyTorch tensor，並padding ，而label中padding的 token id是-100，可使模型不計算padding的loss

3.Train

我的模型訓練使用得超參數都設置為

Batch\_Size = 32, Learning\_rate = 0.002, Epochs = 35

訓練的步驟:

首先清除之前迭代中計算的gradient，將batch中的字典內容作為參數傳遞給模型，之後計算獲取損失值，再利用loss算出gradient，然後更新模型參數

4.Evaluation

下圖是T5/GPT2模型的rouge值。

T5:

一張含有 文字, 螢幕擷取畫面, 樣式, Rectangle 的圖片

自動產生的描述

GPT2:

一張含有 文字, 螢幕擷取畫面, 樣式, Rectangle 的圖片

自動產生的描述

我們可以發現，GPT2的ROUGE-P低於T5，且T5和GPT2在ROUGE-R、ROUGE-F差不多，代表T5比GPT-2在生成的摘要中，有較多参考摘要包含的内容。