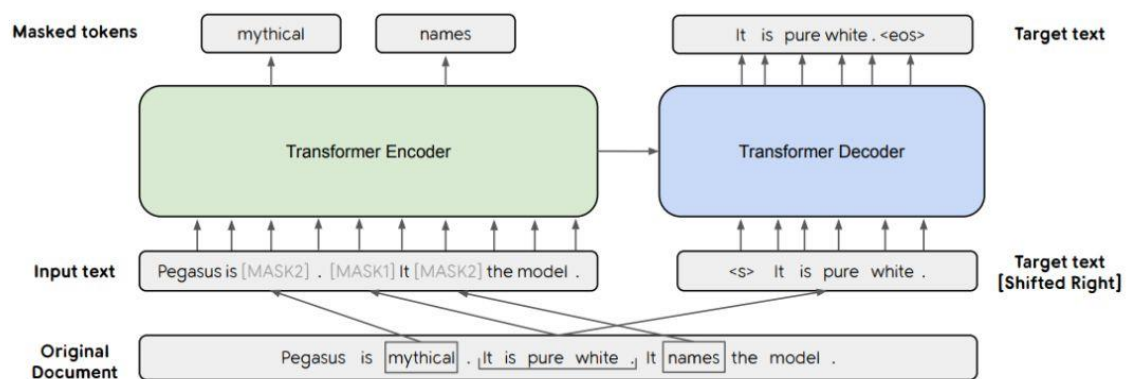


## hw3

### 1.1

這次的 model 是用mt5-small 去fine-tune。是個可以在英文以外的語言進行text generation 的model。這邊找了網路上的圖。

Cited: [中文生成模型T5-Pegasus详解与实践 SunnyGJing的博客-CSDN博客 t5-pegasus](#)



就是 encoder+decoder 的架構，像 transformer 那樣，把文字 tokenize 後當作輸入，經過上面的模型後，在 decoder 的部分會逐一得到當下最有可能的文字的機率分布。然後根據選擇文字的策略去決定要選哪些文字，最後就可以得到一段文字的輸出。經過 training 就可以達成 summarization 的目的。

### 1.2

首先用 jsonlines 套件讀取檔案，用 `AutoModelForSeq2SeqLM.from_pretrain` 下載 mt5-small，用 `load_datasets` 讀取 maintext、title、id，把 maintext 不該有的

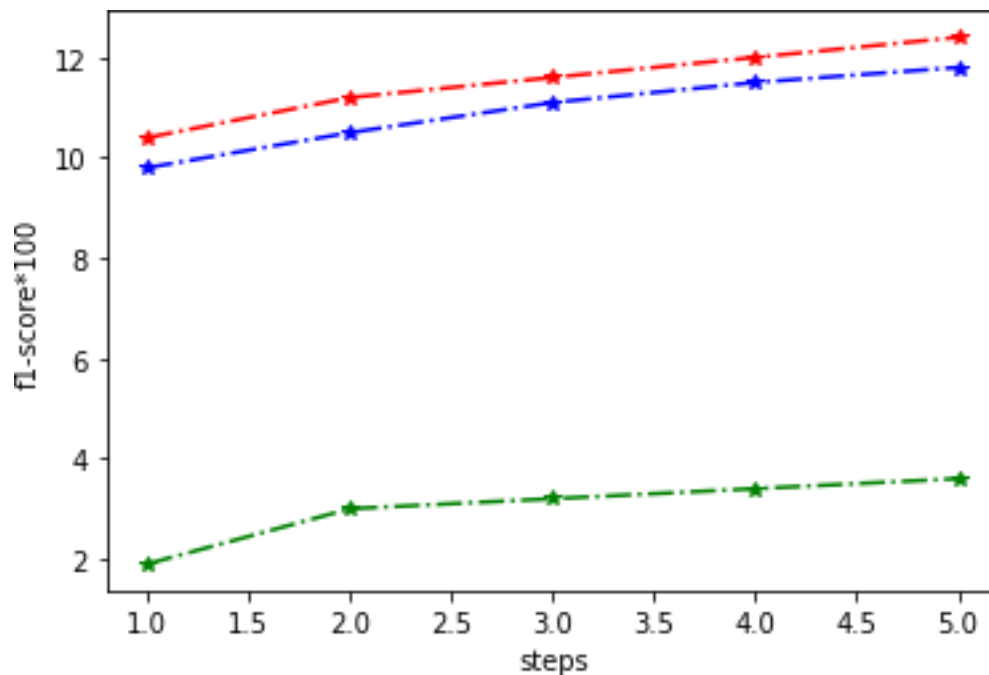
前後\n 和空格去掉，把資料存為datasets 的格式。然後處理資料，maintext 前面加上 summarize: ，因為 T5 模型加上prefix 對於 fine-tuning 會比較好。然後把maintext 跟title 分別丟到 tokenizer 裡面，變成像是bert 那樣的input 形式。然後把 optimizer、learning rate scheduler、dataloader 都設定好就可以開始 train 了。

## 2.1

lr=5e-5、max input length=1024、max output length=128、batch size=2 \* gradient accumulation steps:4 =8，epoch=10。因為我是改寫自 huggingface 的sample script，增加 epoch 數後 train 的效果還不錯，所以就沒特別改寫參數。

## 2.2

以下是在第 1000、2000、3000、4000、5000 個train steps 的 model 的performance。



其中紅色是 rouge-1，藍色是 rouge-1，綠色是 rouge-2。顯然隨著訓練 steps 上升成效也會變好。

### 3.1

這 5 個 strategy 都是在 decode 出 text 對於文字選擇的 strategy

Greedy 就是在所有可能的單字中選機率最高的單字作為下一個單字。但因為都是選機率最高的，所以生成的文本時常會重複。而且因為只關注下一個，不是關注全部，所以整體而言生成的文本的對應機率不一定是最高的。

Beam search 是同時保留當下 n 個機率最高的選項，然後走下去。這樣比較容易選到整體而言機率最高的那段文本。但不保證。如果 n 太大，就可能生成的文本與輸入相關性不高。

Top k 這個方法是做sampling 的一種。Sampling 就是不要單純考慮機率最高，而是隨機從所有可能的單字挑一個。但挑到機率超低的單字也很奇怪，所以 top k 就是從機率最高的 k 個做sampling。這樣可以保證diversity 也可以保證出現的單字不會太奇怪。但如果分布過於不平均，還是有可能會選到機率比較低的單字。如果分布過於平均，就有可能會選不到機率比較高的單字。

Top p 這個方法是做sampling 時，不是考慮機率最高的，而是考慮機率從高到低總合大於p 的subset。這樣可以避免上面 top k所講述的問題。

Temperature 是一種控制機率分布的方法。單字的分布通常是通過 softmax 的，而temperature 可以去調整 softmax 的參數來達到控制機率分布是偏 uniform 還是spiky。

### 3.2

Greedy:

```
{
  "rouge-1": {
    "r": 0.16815073677689402,
    "p": 0.284196176975044,
    "f": 0.19954101962383752
  },
  "rouge-2": {
    "r": 0.0638774492197515,
    "p": 0.1094403243170056,
    "f": 0.07547458485155657
  },
  "rouge-l": {
    "r": 0.1544466878809097,
    "p": 0.261219440534732,
    "f": 0.1832564741112964
  }
}
```

Beam search: num\_beams=5

num\_beams=10

```
{
  "rouge-1": {
    "r": 0.16815073677689402,
    "p": 0.284196176975044,
    "f": 0.19954101962383752
  },
  "rouge-2": {
    "r": 0.0638774492197515,
    "p": 0.1094403243170056,
    "f": 0.07547458485155657
  },
  "rouge-l": {
    "r": 0.1544466878809097,
    "p": 0.261219440534732,
    "f": 0.1832564741112964
  }
}
```

```
{
  "rouge-1": {
    "r": 0.1732970225365819,
    "p": 0.2772376081303498,
    "f": 0.20203119670638997
  },
  "rouge-2": {
    "r": 0.06644041254336534,
    "p": 0.10826319774261141,
    "f": 0.07741968423171317
  },
  "rouge-l": {
    "r": 0.1587462791122639,
    "p": 0.2544692505385697,
    "f": 0.18514601656871563
  }
}
```

看起來 beam 數比較大時，表現會比較好。

Top k=50

```
{
  "rouge-1": {
    "r": 0.12350941930991521,
    "p": 0.1893870753527657,
    "f": 0.1401134832001723
  },
  "rouge-2": {
    "r": 0.03645906335770042,
    "p": 0.05584490469994369,
    "f": 0.04086561984719768
  },
  "rouge-1": {
    "r": 0.1121712795998359,
    "p": 0.17359195934913596,
    "f": 0.12768378362231494
  }
}
```

k=70

```
{
  "rouge-1": {
    "r": 0.11982120283589914,
    "p": 0.17997158375648858,
    "f": 0.13462511024837606
  },
  "rouge-2": {
    "r": 0.03502022758038007,
    "p": 0.05114574127539276,
    "f": 0.03844449546739392
  },
  "rouge-1": {
    "r": 0.10887968276810345,
    "p": 0.16514685721127673,
    "f": 0.12272441182924072
  }
}
```

看起來 k=50 時已經有足夠多的選擇了，放大到 70 反而表現沒那麼好。

Top p=0.9

```
{
  "rouge-1": {
    "r": 0.12872004546881763,
    "p": 0.20476373830507477,
    "f": 0.148021697088809
  },
  "rouge-2": {
    "r": 0.04013130491951559,
    "p": 0.06267914551522083,
    "f": 0.045065135852554346
  },
  "rouge-1": {
    "r": 0.11744265370570278,
    "p": 0.18845808146709733,
    "f": 0.13536765879914514
  }
}
```

p=0.7

```
{
  "rouge-1": {
    "r": 0.13806598446921373,
    "p": 0.2390530624027483,
    "f": 0.1650419584593162
  },
  "rouge-2": {
    "r": 0.04477051718479613,
    "p": 0.0781298540232126,
    "f": 0.05299764556076135
  },
  "rouge-1": {
    "r": 0.12716457798175873,
    "p": 0.2210964181324671,
    "f": 0.15212615117566627
  }
}
```

看起來  $p=0.7$  時比  $p=0.9$  時好，可能是  $p=0.9$  時選擇太多了，所以可能選到一些不是那麼相關的單字，所以表現就沒那麼好。

Temperature=0.9

=0.7

```
2022-11-27 09:29:18.920979. E t
{
  "rouge-1": {
    "r": 0.13271908684151257,
    "p": 0.2180636357401144,
    "f": 0.15500130930630057
  },
  "rouge-2": {
    "r": 0.04142034194634412,
    "p": 0.0667197885181343,
    "f": 0.04750842727396011
  },
  "rouge-l": {
    "r": 0.12202614782343041,
    "p": 0.2014464684401783,
    "f": 0.14266258795290593
  }
}

{
  "rouge-1": {
    "r": 0.13752728997597577,
    "p": 0.23319677726593244,
    "f": 0.16294398990060946
  },
  "rouge-2": {
    "r": 0.044504262857560394,
    "p": 0.07466692293910682,
    "f": 0.051926697877940076
  },
  "rouge-l": {
    "r": 0.12629452766456528,
    "p": 0.21492646750193745,
    "f": 0.14971269150551395
  }
}
```

看起來是 0.7 的 temperature 的表現比較好。

以上所有的 case 中，beam=10 的表現最好，所以我決定用 beam=10 當作這個作業的結果。