國立臺灣大學 National Taiwan University

指導教授：陳縕儂
學生：黃泰誠
學號：R09921083

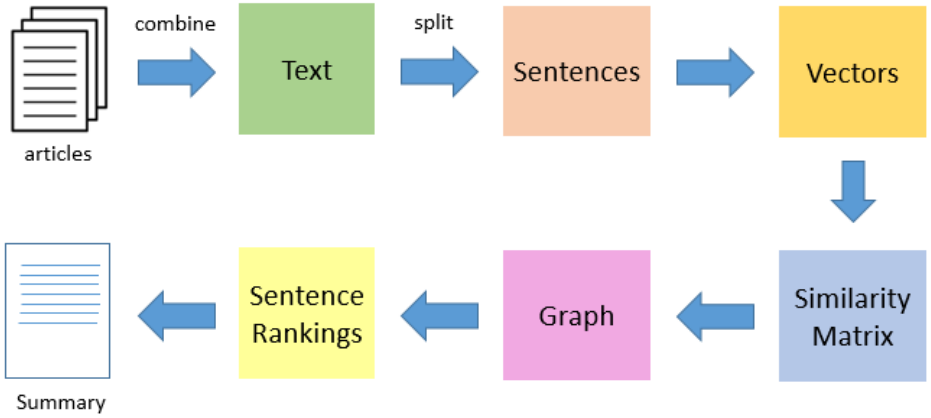
Applied Deep Learning Homework 3

Q1:Model

Model : MT5 – small

Mt5 即 Multilingual T5，T5 的多國語言版，出自最近的论文 mT5: A massively multilingual pre-trained text-to-text transformer 訓練和 bert 相似只不過使用了 seq2seq 的版本，它使用了標準的 Encoder- Decoder 模型， 並建構了無監督/有監督的文本生成預訓練任務。

在做 summarization 需先將資料 encoder 等訓練完成後需要再將資料進行 decoder，首先會使用 special token encoder 從一個文章中找出相關性較高的句子來做為輸入，透過 mt5 的 model encoder 可以找出相似率較高的一些關鍵字作為輸出，之後再根據 decoder 所採用的 strategy 來產生出文章的 summarization ，如下圖所示。



Preprocessing:

首先我先將課堂給的.jsonL 檔案轉成.json 檔，再將其丟入 model，再資料送入 model 之前需要先進行 tokenization 然而根據 mt5-small 的 vocab 我將每個中文或是英文字轉成數字，而 vocab size 的大小為 250112，輸入的訓練資料有 title,maintext,這裡的 title 會直接作為 labels 來做訓練，型式如下圖所示。

Maintext:

```
'input_ids': [259, 1083, 103194, 105763, 107812, 161430, 82813, 73776, 38970, 29630, 7605, 493, 86248, 2371, 96270, 261, 6124, 4768, 35775, 4002, 460, 31389, 65262, 60097, 4280, 18643, 24322, 306, 155113, 6892, 312,
```

Labels:

```
'labels': [259, 161430, 275, 2188, 9139, 82813, 73776, 38970, 29630, 6124, 4768, 35775, 143296, 65262, 60097, 4280, 190686, 162536, 1]
```

Q2: Training

Hyperparameter:

Learning rate: 3*10-5

選擇原因:雖然曾經試過用 5*10-5 但是準確率沒有 3*10-5 來得好所以想說應該是 summarization task 的 learning rate 不能夠太大

Epoach : 20

選擇原因:再一開始的時候設置成 10 但是最後發現在訓練到後期 accuracy 依然有在上升所以調成了 20。

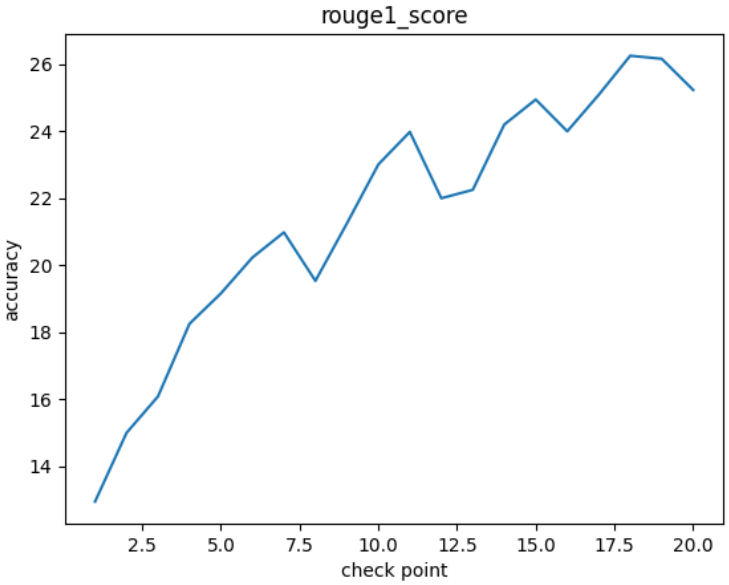
Batch size:1

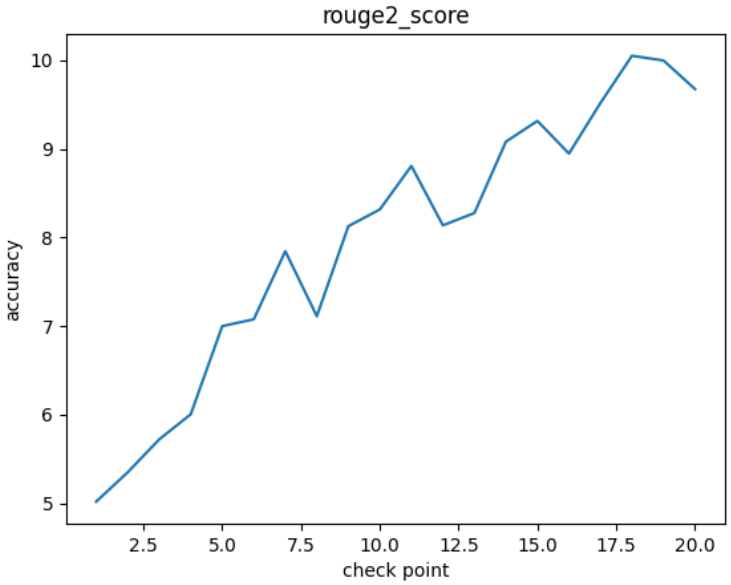
選擇原因:記憶體容量不夠

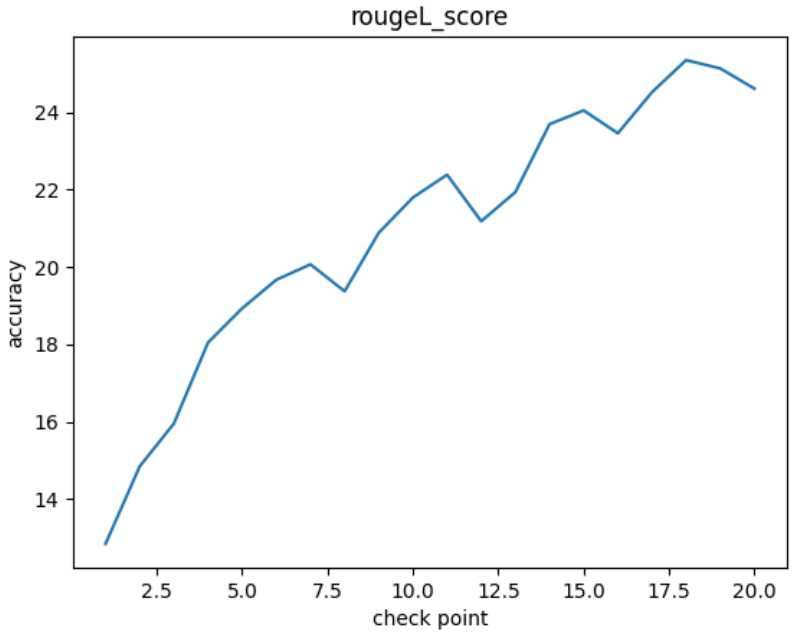
Gradient accumulate :8

選擇原因:能夠避免 loss 的震盪

Learning Curves:







Q3: Generation Strategies

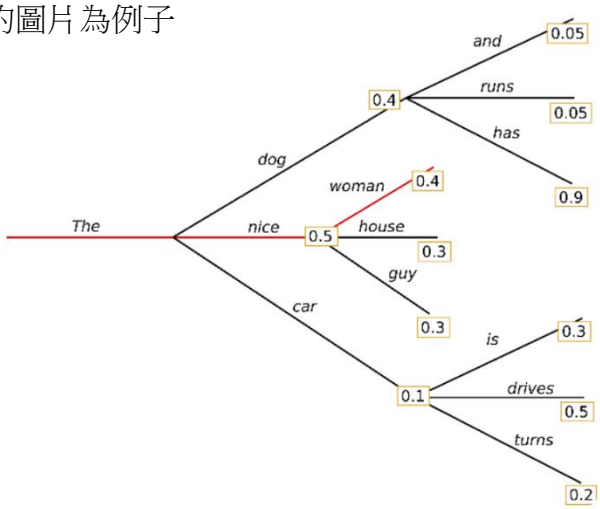
Stratgies:

- **Greedy**

  Greedy search 只選擇概率最高的單字作為其下一個單詞

  $$wt = \text{argmax}_w P(w \mid w1{:}t{-}1)$$

  可以看到以下的圖片為例子



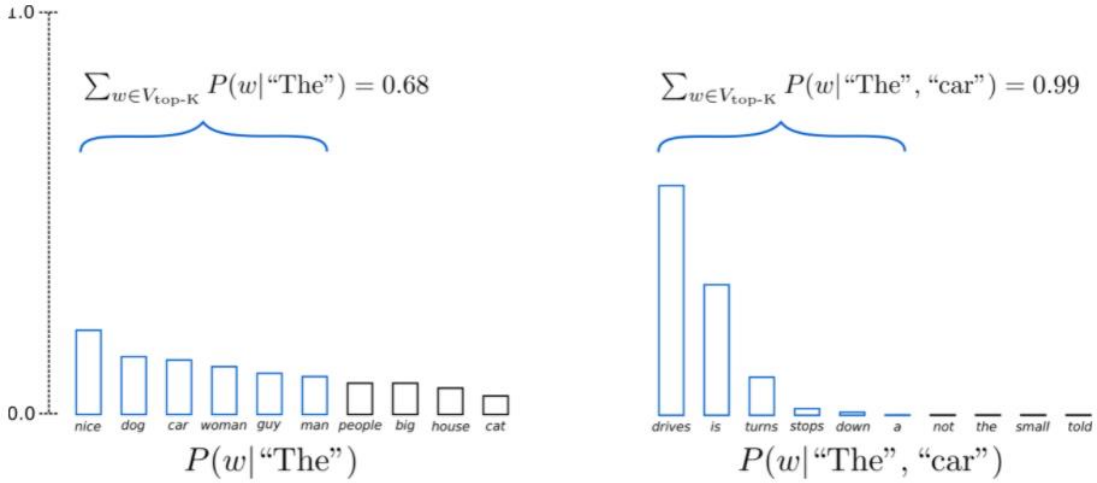
  從”The”這個字開始，greedy search 就直接選擇了 nice 做為下一個字的預測，因為 nice 的機率為 0.5 相較於其他選項是來的最高，根據這個原則最後選擇了 (“The”,”nice”,”woman)，然而這條路的機率相乘起來只有 0.5*0.4 = 0.2，但是如果選擇(“The”,”dog”,”has)那這條路的機率就是 0.4*0.9 = 0.36，明顯的比較高，所以 greedy search 會有一步錯步步錯的問題存在。

- **Beam search**

  Beam search 解決了上述的缺點，這個策略會選擇總體概率最高的假設，這樣就可以降低丟失隱藏的高概率單詞序列的風險，所以使用 beam search 的話效果上會比 greedy 好，然而 beam search 有一些參數可以去調整，舉例來說 num_beams 的設定可以選擇下一個單字可以有多少選擇，如果就上面那張圖片所示的話 num_beams = 3，除此之外還有 no_repeat_ngram_size 這個參數的調整可以決定我們在 decode 的時候要讓同一個字詞最多出現幾次，再來是 num_return_sequences 的參數設定了我們要在最後的輸出中 predict 多少句子來作為我們輸出的參考，然而上述的這些參數都得經過多次的嘗試才能找到適合的數值，可以說 beam search 是個可以加以研究的策略。

- **Top-k Sampling**

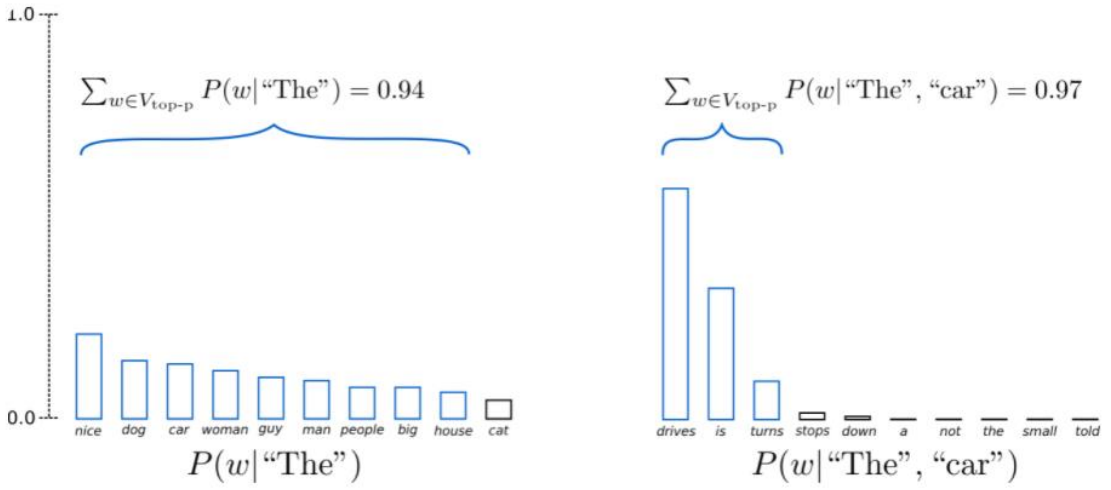
  Top-k 是個簡單但是非常有用的策略，在 Top-k 的取樣中我們可以選擇 k 與前一個字比較下來機率最高的字之後再對這 k 個字做選擇，這邊假設我們原先有 10 個單字而我們的 k 設定為 6:



  這樣一來就可以過濾掉(“not”,”the”,”small”,”told”)這幾個比較不相關的字。

- **Top-p Sampling**

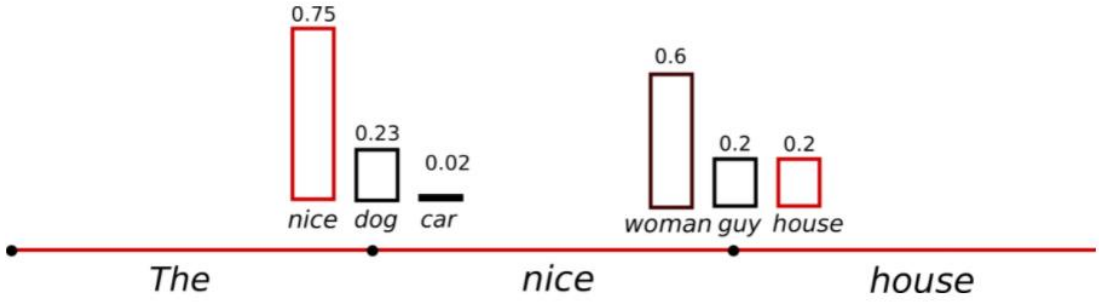
  Top-p sampling 是累積前幾個最可能出現的單字來加總，累積超過機率 p 的單字中進行選擇。然後在這組詞之間重新分配機率。這樣一來詞集的數量可以根據下一個詞的機率分佈動態增加和減少，下圖為範例的解釋



  假設 p=0.92，Top-p sampling 選擇了加總起來超過 0.92 但又是最接近 0.92 的所有字也就是 0.94，不過我們發現當我們選擇了第一個字為比較是屬於比較沒有辨識度的，那麼下一個字就沒辦法過濾太多選項，例如 P(w｜"The”)的機率分佈比較平均，所以只能過濾掉一個選項，但是如果是 P(w｜"The","car")就可以過濾掉六個選項。

- **Temperature**

  Temperature 可以讓機率分佈 P(w|w1:t−1)更精準，透過增加高機率的單字的可能性並降低低機率單字的可能性。



在第一步的時候就可以明顯去除了(“car”)的選項。

Hyperparameters:

- **Greedy**

Num_beam = 1

準確率


```
{
  "rouge-1": {
    "f": 0.2520715244620719,
    "p": 0.2707370904504384,
    "r": 0.24854774492365875
  },
  "rouge-2": {
    "f": 0.09779377411534647,
    "p": 0.10471543352538992,
    "r": 0.09695066067296243
  },
  "rouge-l": {
    "f": 0.23348688940530118,
    "p": 0.25988579130875333,
    "r": 0.22320028074003037
  }
}
```


num_beam = 1

no_repeat_ngram_size = 2

準確率


```
{
  "rouge-1": {
    "f": 0.25438116572818825,
    "p": 0.27324801299933715,
    "r": 0.2505098846346536
  },
  "rouge-2": {
    "f": 0.09587726449437661,
    "p": 0.10282587779318682,
    "r": 0.09481885437475487
  },
  "rouge-l": {
    "f": 0.2274381401109118,
    "p": 0.24597199227705652,
    "r": 0.2220086187972218
  }
}
```


Num_beams>1 就進入 beamsearch 的模式了

■ **Beam search**

num_beams = 3
no_repeat_ngram_size = 3
num_return_sequences = 3(取 0)
準確率


```
{
  "rouge-1": {
    "f": 0.2676162724181281,
    "p": 0.28410609423052063,
    "r": 0.26654404657077
  },
  "rouge-2": {
    "f": 0.10964281345908324,
    "p": 0.11703364061855705,
    "r": 0.10917103093999066
  },
  "rouge-l": {
    "f": 0.24132317359804162,
    "p": 0.2582353004903973,
    "r": 0.23816764439298557
  }
}
```


num_beams = 3
no_repeat_ngram_size = 2
num_return_sequences = 3(取 2)
準確率


```
{
  "rouge-1": {
    "f": 0.2631310617422328,
    "p": 0.2721293041623207,
    "r": 0.2686063466996996
  },
  "rouge-2": {
    "f": 0.10307399738311243,
    "p": 0.10637263557950688,
    "r": 0.10568478580721889
  },
  "rouge-l": {
    "f": 0.23329171492836717,
    "p": 0.24194298446090703,
    "r": 0.23700059230279075
  }
}
```


備註: num_beams 超過 3 就無法在一小時內完成 prediction，故此參數不做更動。

■ **Top-k Sampling**

Top-k = 10

準確率


```
{
  "rouge-1": {
    "f": 0.2596130130362389,
    "p": 0.27323213487894643,
    "r": 0.2602876503310123
  },
  "rouge-2": {
    "f": 0.10401209905626055,
    "p": 0.10952823995012183,
    "r": 0.1044913901328487
  },
  "rouge-l": {
    "f": 0.23917462984502386,
    "p": 0.2595420743772049,
    "r": 0.23329519786626982
  }
}
```


Top-k = 10
no_repeat_ngram_size = 2
準確率


```
{
  "rouge-1": {
    "f": 0.2625260895839818,
    "p": 0.2783812482246368,
    "r": 0.26124938892081817
  },
  "rouge-2": {
    "f": 0.10369143875381279,
    "p": 0.11003109429632356,
    "r": 0.10324667318417517
  },
  "rouge-l": {
    "f": 0.2336057023552353,
    "p": 0.24818469776993174,
    "r": 0.23163698876677513
  }
}
```

■ **Top-p Sampling**

Top-p = 0.8

準確率


```
{
  "rouge-1": {
    "f": 0.2584670288544965,
    "p": 0.2711618662318313,
    "r": 0.2601529384611409
  },
  "rouge-2": {
    "f": 0.10391872418704265,
    "p": 0.10919385169591336,
    "r": 0.10492681288537749
  },
  "rouge-l": {
    "f": 0.2380699479582984,
    "p": 0.25783773140394806,
    "r": 0.23290605670513456
  }
}
```


Top-p = 0.8

Top-k = 10

No_repeat_ngram_size = 2


```
{
  "rouge-1": {
    "f": 0.2605544625067976,
    "p": 0.2760617685347837,
    "r": 0.2597340165121301
  },
  "rouge-2": {
    "f": 0.1020973961273425,
    "p": 0.1082277059908542,
    "r": 0.10195842725018561
  },
  "rouge-l": {
    "f": 0.23232954692191274,
    "p": 0.24670702521075477,
    "r": 0.23059796298834728
  }
}
```


■ **Temperature**

Top-k = 10

temperature = 0.7

準確率


```
{
  "rouge-1": {
    "f": 0.25361448682492116,
    "p": 0.2843761658264447,
    "r": 0.24037119422995684
  },
  "rouge-2": {
    "f": 0.10018225870378605,
    "p": 0.11192208865280528,
    "r": 0.09544267410766437
  },
  "rouge-l": {
    "f": 0.23380219113899653,
    "p": 0.26926964465001413,
    "r": 0.21679238667573947
  }
}
```


Top-k = 10

temperature = 0.6

準確率


```
{
  "rouge-1": {
    "f": 0.2526789874364545,
    "p": 0.28281947003929864,
    "r": 0.2405175855218088
  },
  "rouge-2": {
    "f": 0.09903274834700088,
    "p": 0.11045570393603392,
    "r": 0.09476711389364374
  },
  "rouge-l": {
    "f": 0.23306263364727156,
    "p": 0.26799995192022025,
    "r": 0.21703245500347723
  }
}
```

My final generation strategy

- **Beam search**

  num_beams = 3

  no_repeat_ngram_size = 3

  num_return_sequences = 3(取 0)