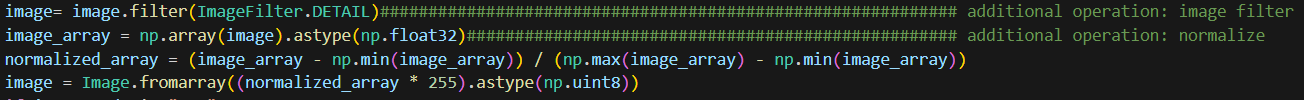
1. Data preprocessing analysis

What adjustment have you done to the Chest X-ray Images or the Reports? (20 pts)

Have you performed any additional operations to filter out noise or extract key information? (10 pts)

針對report進行tokenize、對image套用強化細節的filter後再進行normalize





1. Model & Training Method (20 pts)

What models do you choose? Why? (Please introduce the visual model and language model respectively.) (10 pts)

Encoder使用的是被肺炎X光照數據集finetune過的Vit(vision transformer)(sample code使用的模型)

使用這個模型的原因是因為考量到任務，使用類似資料訓練過的模型可以降低訓練成本。

Decoder 則是Bert模型(case sensitive)

會使用case sensitive版本是因為考量到report的格式，如果模型能考慮大小會比較好。

6/20報告老師有提到bert無法生成，實際上我們有考慮過這個問題，所以有深挖過huggingface的doucumentation，我們發現無論是vision encoder decoder 還是正常的 bert model from pretrained又或是openai gpt2 from pretrained這些huggingface提供的框架其實都有generation的method可以呼叫，所以我們推測實際上應該是當初模型的開發團隊跟huggingface有說好這些基於transformer框架的模型都要有text generation的能力，至於bert的開發團隊是如何攻克bert本身的限制，我們因為無法確認bert的實作方法無法繼續深挖下去。但是可以確定huggingface提供的bert的確可以進行text generation。

Your evaluation scores during training. (10 pts)



1. Analysis (30 pts)

Have you encountered any difficulties? How did you address them? Unlimited (10 pts)

**問題一:**

一開始本來打算使用LoRA來微調，但是hugging face 的 LoRA method不支援vision encoder decoder model。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

**問題二:**

在將tokenizer換成bert專用的tokenizer時因為bos token的id設定不好( bos\_token\_id = 627)導致生出來的text多了”##”

['##Chest PA view shows : Impression : - Suspicious pulmonary ed']

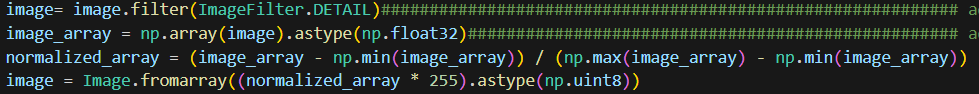
['##Chest PA view : Impression : - Increased both lung markings.']

['##Chest AP view showed : 1. s / p sternotomy and cardiac']

Solution: 將bos\_token\_id設定成400404

**問題三:**

在前處理的階段因為使用的API是pillow，所以最好透過pillow內建的工具完成想做的前處理

Solution:

**問題四:**

Vision encoder decoder model在generate時需要設定bos token id(sample code 沒有這個問題)

Solution: 研究過後發現vision encoder decoder model的config有兩種:

model.config 、 model.generation\_config。在這裡是因為generation config的部分沒有設定所以出問題。



備註:也許openai的gpt2有預先設置這個參數，bert沒有所以出現了這個問題

Sample code的rouge socre這麼慘也有可能是因為這樣，因為model.config跟model.generation\_config的參數不一樣，導致loss有在降，但是eval還是不好。

為了驗證這個的想法，在sample code中加入對generation\_config的設置。

result: rouge-L-P rouge-L-R rouge-L-F rouge-2-P rouge-2-R rouge-2-F



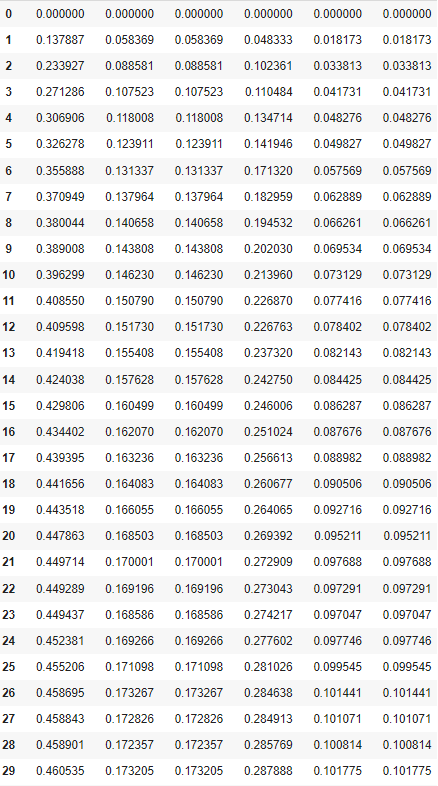
sample code:

rouge-L-P rouge-L-R rouge-L-F rouge-2-P rouge-2-R rouge-2-F

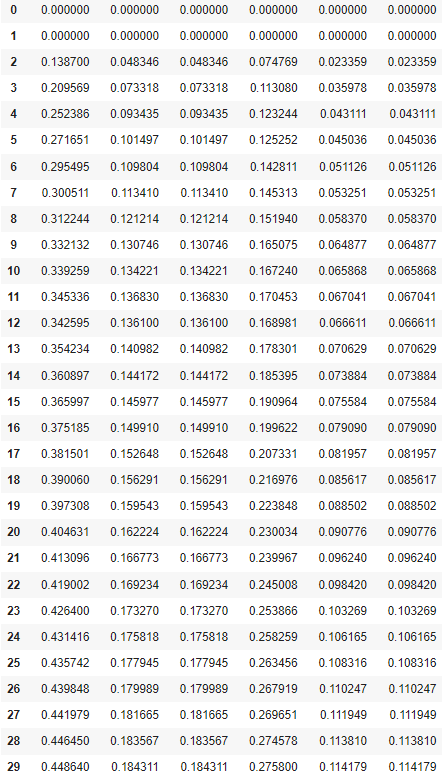
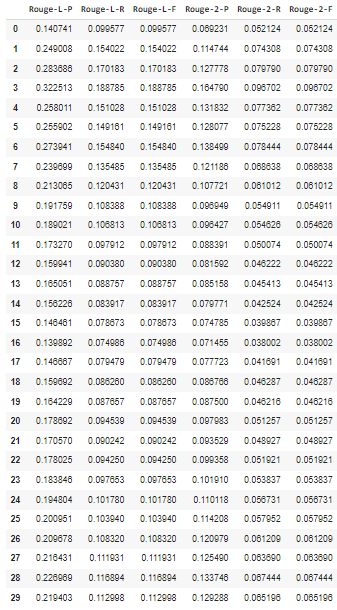
0.275969, 0.058301, 0.058301, 0.098075, 0.016161 , 0.016161

可以看到結果有稍微變好。

**問題五:Rouge分數再提高**

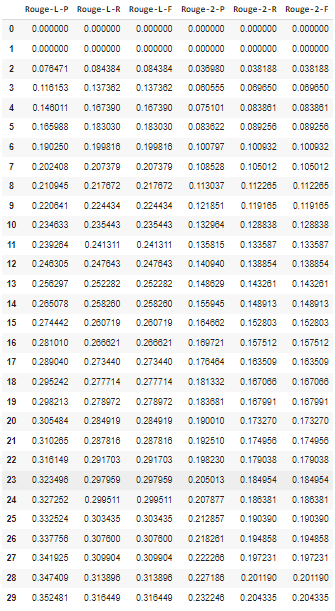
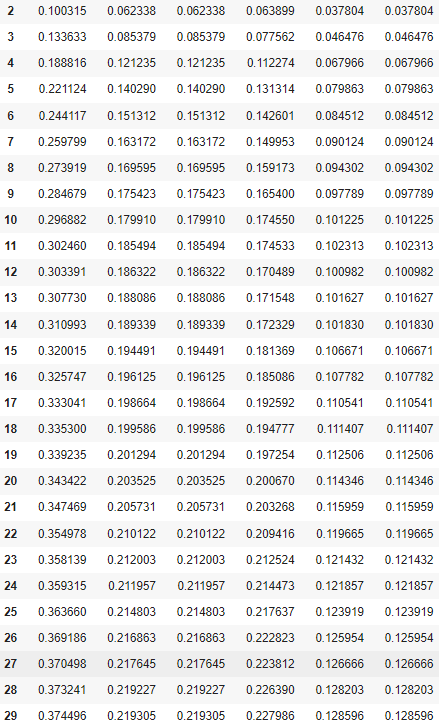
在前面的部分中，我們已經讓rouge分數提高了許多，在ROUGE-L-P，以及ROUGE-2-P都取得了蠻不錯的分數，但是在剩下的幾個分數都偏低，而這有可能是因為，生成最大長度限定在20的關係，實際上所對到的LCS以及二元組其實都沒有到非常多，僅是因為分母小所以被拉高了，這點也可以在後面的ROUGE-L-R和ROUGE-2-R中得到證實，因此我們針對這一部分進行了調整，首先我們將max\_length更改為30，而得到的結果如下

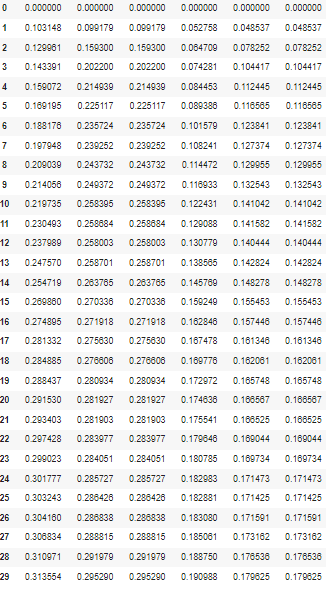
雖然在準確率的方面有略微下降，但是後面的召回率不管是ROUGE-L還是ROUGE-2都有明顯的上升，表示實際上的結果有所改善，而在這之後，我們又嘗試了在訓練時以及實際測試時不同長度的作法，分別是訓練時max\_length=30，測試時max\_length=40，還有訓練時max\_length=30，測試時max\_length=40，而結果分別如下



而上述兩種的結果，在第一種時雖然前面分數有較快的成長，但是隨後分數卻下降許多，甚至在最後的結果相較原本max\_length=20時差了不少，但在第二種情況，表現甚至接近均設定為30時，而這可能是因為，訓練時雖然只設定了長度30，但後面測試時變為40，不僅將訓練的部分完整用出，甚至後面多的一段也多少有對中一部分內容，且模型在生成時也未必要達到最大長度，而前期分數進步較慢也是因為還沒把控好長度，相反的，訓練時比測試時設定的最大長度還要長，可能會導致部分訓練的重點內容沒有辦法表現出來，進而導致分數下降，也有可能是輸出最後段的內容表現較好，被截掉導致分數下降，使得雖然後面有再回升，但也無法達到跟之前相同程度

而在這之後，我們又分別將max\_length設為50 100 150並進行了測試，而得到的結果如下





在上面的三種結果中，我們可以看到max\_length在100時表現是全部裡面最好的，而max length從30到50，以及50到100時，都可以看到最後結果的改善，但是到了150時，卻反而下降，而且我們也可以看到，在max\_length為100時，不管是在rouge-L，還是rouge-2的分數，準確率跟召回率都非常的接近，表示這時候可能生成的長度也比較接近，而到了150時，雖然也相當接近，但分數卻都較100時低，表示max\_length到了一定的程度之後，都會自己去偏向和原報告接近的字數，但表現上卻不一定會更好，超過一個程度反而會下降，而還有一點可看的是，進行訓練所需要的時間，在max\_length是30的時候，訓練僅需11分鐘，但到了100時，訓練需要23分鐘，而150時更到了27分鐘，表示max\_length愈長所需的訓練時間也會越長，在緊緊訓練了30epoch的情況下就有了明顯的差距，若需要進行大量訓練，如何權衡分數和時間也是一個課題

**備註: Rouge test 的程式碼保有在 rouge\_score\_test.ipynb中、其生成解結果保有在rouge\_score\_test\_generated\_output.csv中**

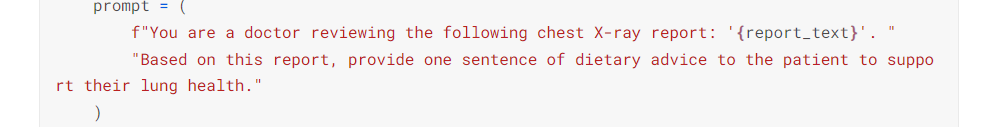
Proposal:

胸腔食療助手: 根據你的胸腔X光照提供飲食建議的幫手。

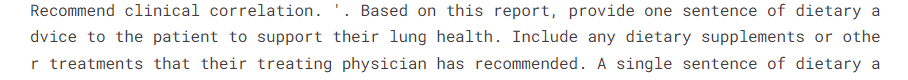
**implementation 方向1 : 使用gpt2 API**

嘗試使用gpt2，將report輸入到模型，並且給予一個prompt，請gpt2扮演醫生的角色生成飲食建議，但是最後輸出的結果並不好，準確率很低，常常回答得不知所以然。(我們認為是GPT2本身能力有限)

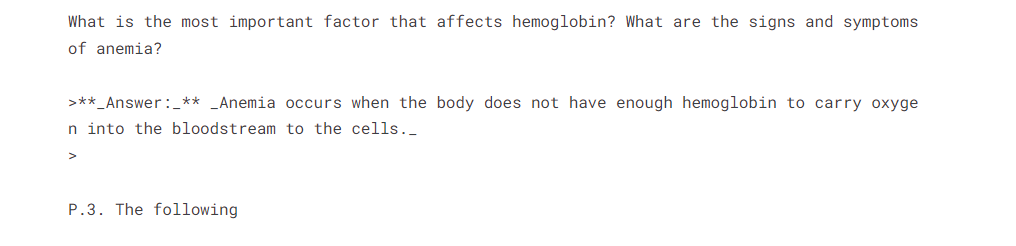
* prompt



* gpt2生成出來的結果(勉強可以的)



* 不準確的輸出



原本想使用gpt3模型，但是gpt3模型需要付費，所以選擇用ChatGPT手動生成建議，再自行做訓練。

**(方向1: 失敗)**

**implementation 方向2 : 利用GPT生成dataset再比照sample code方法訓練**

**ChatGPT**

dataset : 將助教提供的dataset的report給chatGPT生成飲食建議

給ChatGPT的提示詞:

you are a doctor. now you are going to give advice to a patient based on provided chest x-ray report i am going to provide you. please conduct your opinion in one sentence. your advice should focus on diet.

(有刻意避免summarize的字樣防止結果不夠口語化)

訓練方法-方向2:

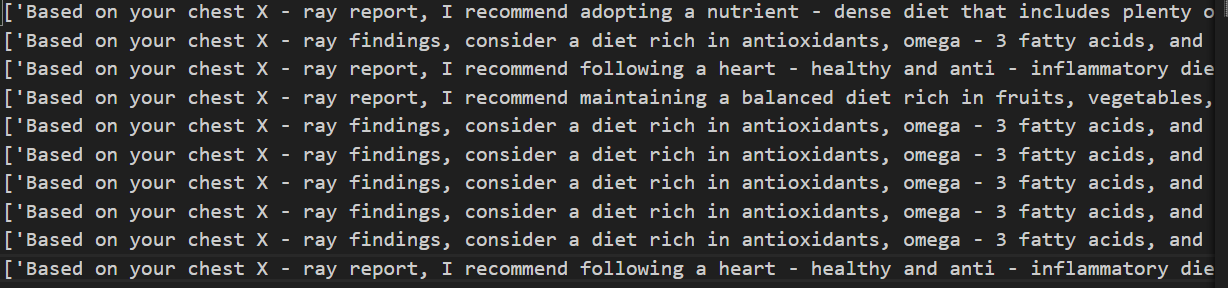
與sample code一致。

model: 與先前提到的一致: vision encoder decoder

encoder: vit 、 deocder: bert

結果:

跑了evaluation dataset中的圖片後:



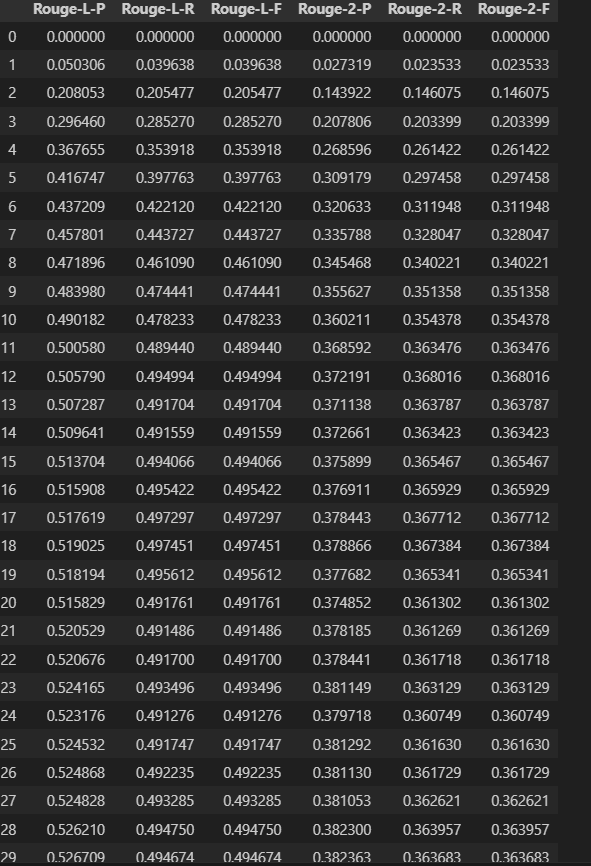
會這樣的原因是因為一開始的report有相當一部分是健康的人的，因此他們的report也差不多結果是我們最後做好的dataset有一部分長得差不多，才有這樣的結果。

**備註: Proposal保有在proposal.ipynb中、**

**其dataset保有在data/proposal\_train & data/proposal\_valid中**

**生成結果保有在proposal\_generated\_result.csv中**

rouge score:



分數會變成這樣主要是因為有特別去調整長度讓output跟target text長度差距不要過大。除此之外還有因為GPT給的建議開頭也長的相似，另外還有一部分的飲食建議其實是相似的所導致。

**備註: 這裡rouge target的text來源於助教提供的validation dataset report經過上面我們生proposal需要的dataset一樣的流程生出來的(一樣的提示詞)。**