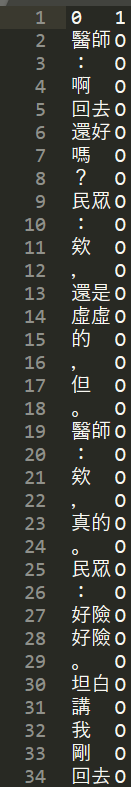
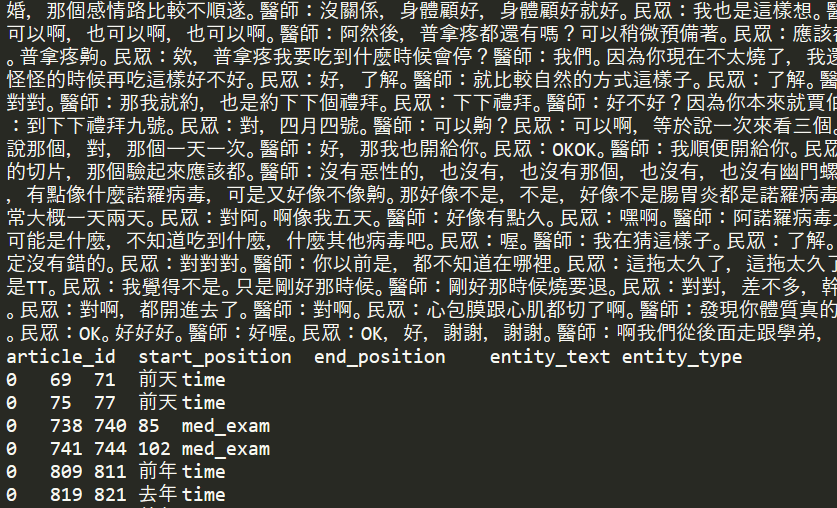
流程說明 :

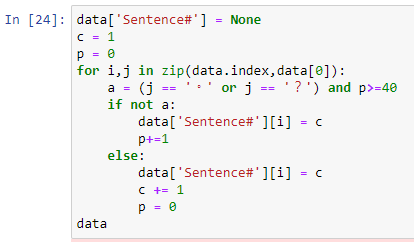
前處理 : 對話轉成NER格式

File Name : baseline.ipynb & Data\_processing\_ex.ipynb

先利用官方提供的baseline.ipynb將資料轉成NER的格式

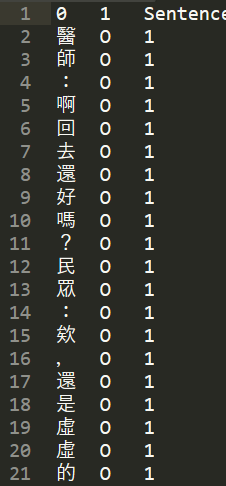


這裡我們使用 Data\_processing\_ex.ipynb 決定每個句子的長度



將每個句子長度控制在40上下 (實測過後這個range對我們的模型效益最好)

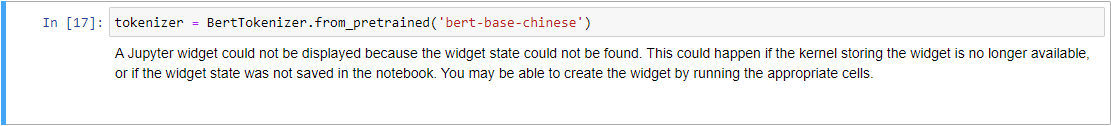
整理完後，文件將呈現成這樣 : (訓練集與測試集都是這樣)



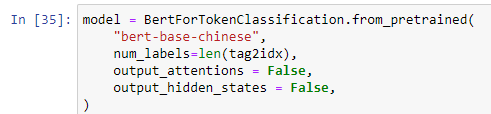
訓練過程 : 訓練NER格式的資料集

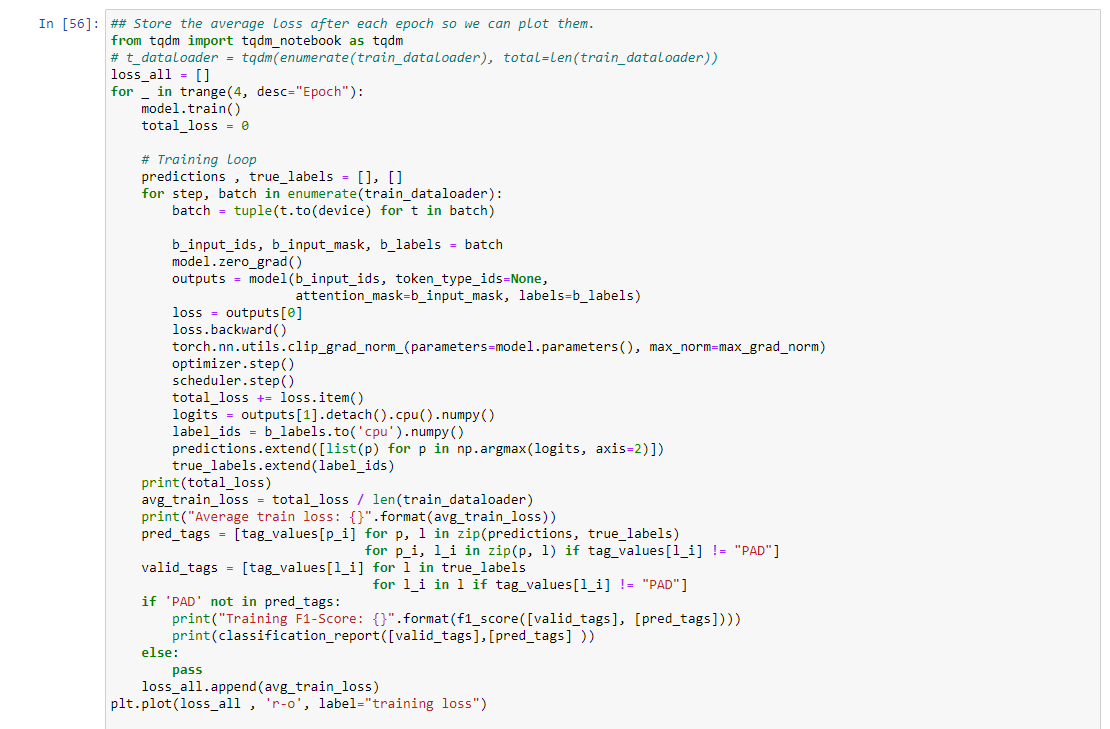
File Name : Bert.ipynb

訓練過程我們使用bert-fine tune，這也是為什麼我們使用字元作為我們的切詞格式(原本我們使用ckip)，因為bert-chinese-base是以字元做為tokenize以及模型的embedding(我們也嘗試過roberta)

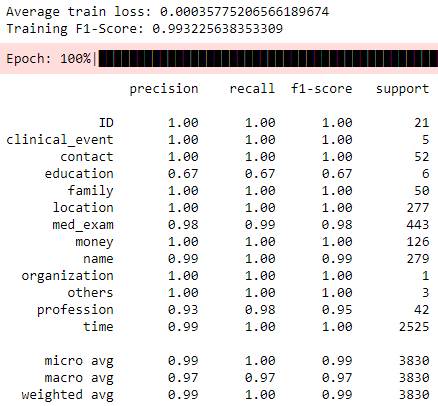






以上為training過程

超參數(最好的那一次)

Epoch : 50

Max\_norm : 2.9

Batch\_size : 8

Sentence\_padding : 500

Optimizer : AdamW

Learning-Rate : 3e-5

訓練過後，我們會做兩件事情，一個是將模型預測測試集的結果輸出，另一個是會檢查在訓練集中的混淆矩陣，而我們發現FN特別多，尤其是time類別的



這時候就需要用後處理將time的FN拉高，不過此舉動會讓FP提高進而導致Precision下降，不過整體來說分數是會上升的

正規表示法後處理 :

File Name : Post\_Process.ipynb

我們會利用正規表示法 : 把我們在檢查FN的地方不足的部分將其補上，例如[月|年]底這種類別，完整的類別可以看Post\_Process.ipynb，補了不少項目，補完後再輸出成Aidea上傳格式