**引人注目的標題生產器**

資管四 黃啟宏

資管四 劉品枘

資管四 李翊辰

資管四 許亦佑

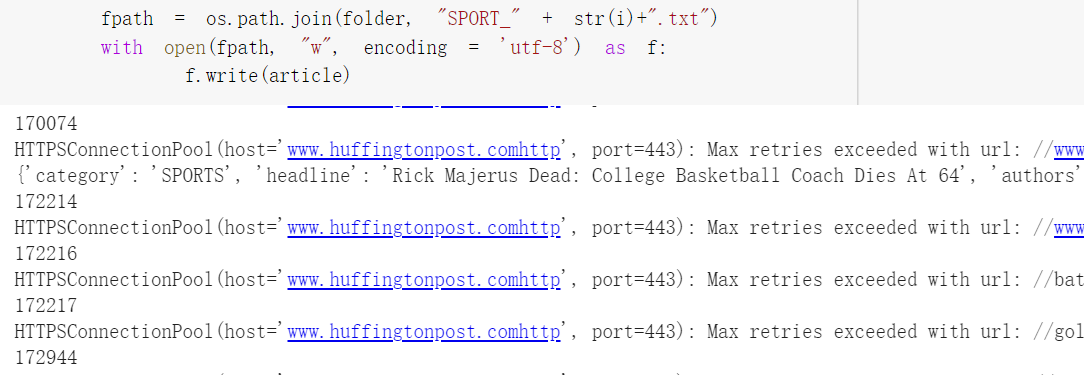
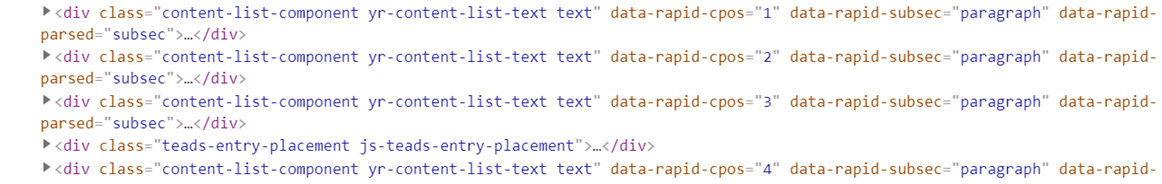
**Abstract**

在這農場標題滿街跑的時代，如何自動化的產生出一個引人注目也同時跟內文切合的標題便成為了一個值得探討、研究的議題，而我們在資訊檢索與文字探勘導論這門課的期末專案，便是選擇這個作為我們的題目。

而在我們的初步構想中，我們打算利用兩種方法來實踐，第一種是利用課程中教授教導過的n-gram以及markov chain建立Language Model以生成最終的文章標題；第二種則是使用Deep Learning中的Seq2Seq Model來訓練出一個文章標題生產的Model。

但在進行模型訓練前，我們必須先經過爬蟲的動作得到資料集，接著再對資料進行一些前處理，但要注意的是在前處理的過程中，我們會針對這次的主題而採取一些前處理的改變。

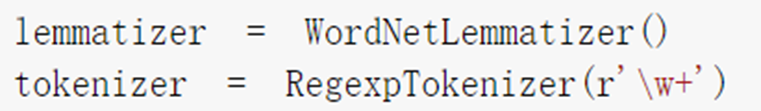
**Crawling**

****

**Preprocess**

Normalization and Lemmatization

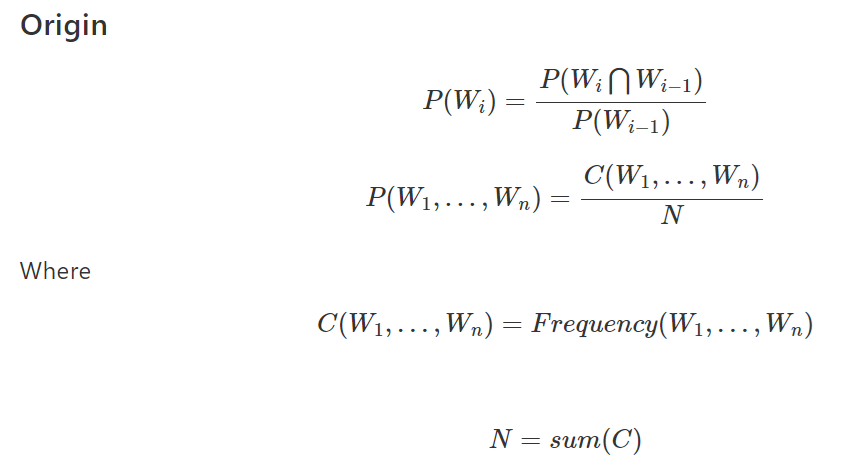
但為了生成符合文法的標題，要特別保留Stop words

nltk 

最後存成Pickle檔

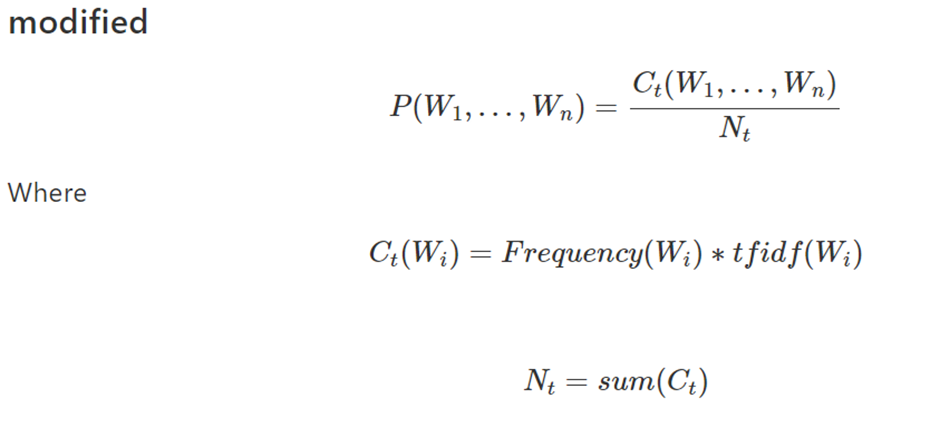
**Bi-gram Model**

在上課時，老師有稍微提到利用馬爾可夫鏈建構出較為單純的機率模型n-gram模型。從下圖可以看到，原本的機率為如下:



利用出現次數總數為分母，各其餘出現次數為分子計算出各詞彙出現的機率，再利用random choice依機率選取。

在我們的實作中，想要更加凸顯N-gram的重要性，因此我們利用出現次數乘上權重(tf-idf)作為改良過後的機率模型，如下:



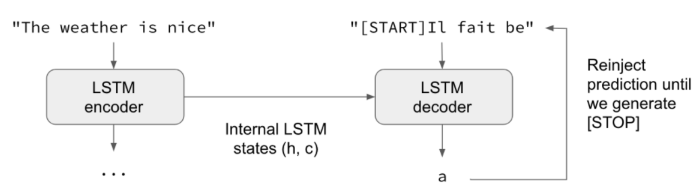
在此方法中，輸入為文章內容和欲生成的標題長度，利用改良後的n-gram模型生成一份標題文字，步驟如下:

1. 找出此文章tf-idf最大的term，作為文章的開頭
2. 利用改良後的n-gram，依標題長度生成i次後面接續的文字
3. 得出初使結果後，利用NNsplit將標題斷句斷在語意通順處
4. 從農場文集合隨機選取一篇，加入引人注目的詞句

在測試過後，我們發現**3-gram結果為最好的**。

程式碼內容見於N-gram.ipynb

**Seq2Seq Model**

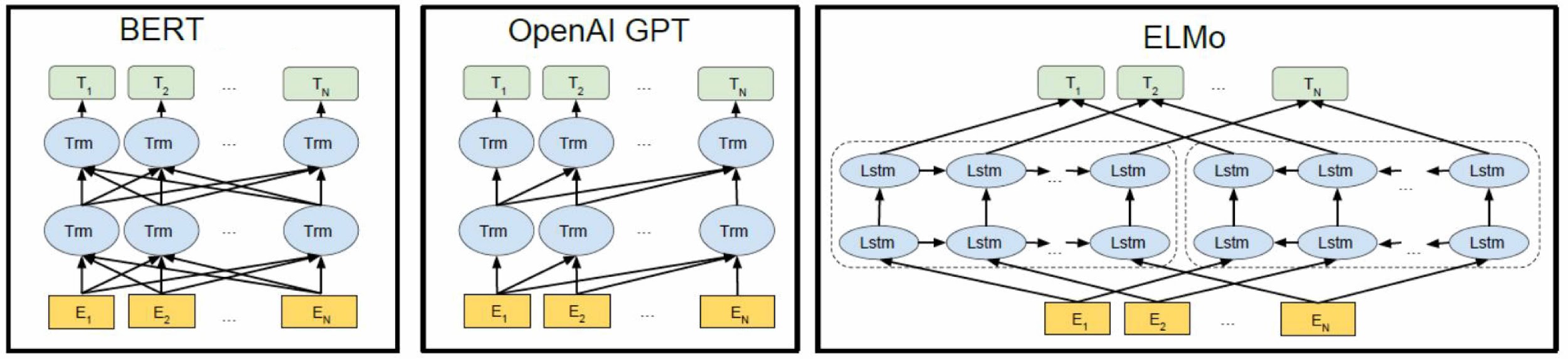
****

使用深度學習中的seq2seq model，將一串文字丟入LSTM/GRU Encoder，然後Encoder的output再丟入另一個LSTM/GRU 的 Decoder 還原成指定大小的詞向量。

Transformer有三種Encoder：

1. BERT (Biderectional Encoder Representation from Transformer)

2. GPT (Generative Pre-trained Transformer)

相較於LSTM或是ELMo (Embeddings from Language Models，一種 BiLSTM)，Transformer使用了一種十分強大的方法 -- Self-Attention 機制。

簡單來說，Encoder 讀入輸入數據，利用層層疊加的Self-Attention 機制對每一個詞得到新的考慮了上下文訊息的表徵。Decoder 也利用類似的Self-Attention機制，但它不僅僅看之前產生的輸出的文字，而且還要attend encoder的輸出。

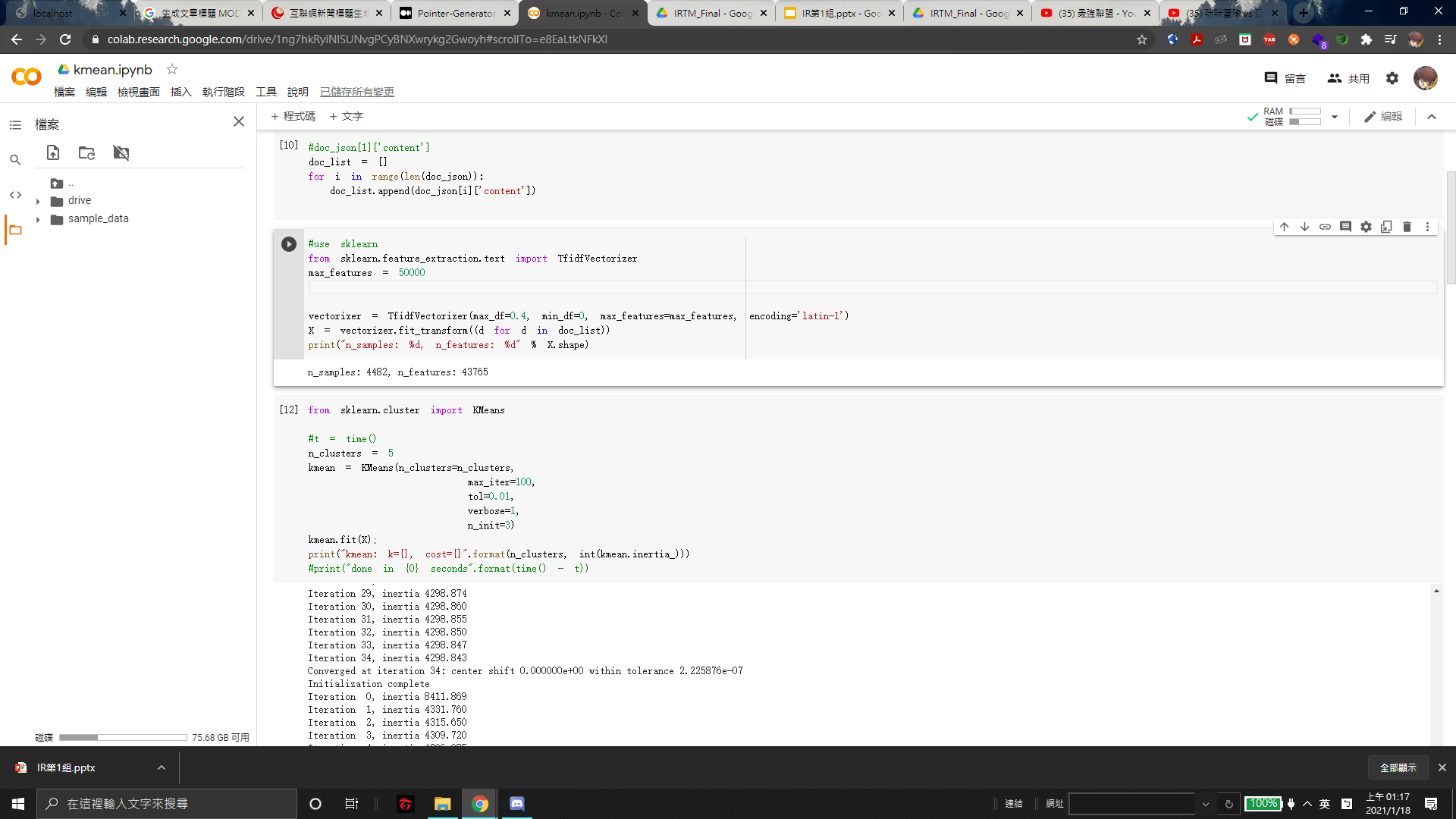
我們以資料集中的 headline 作為 ground true，content作為 training data，然後使用已經 pretrain 好的 model 套件 -- Headliner。

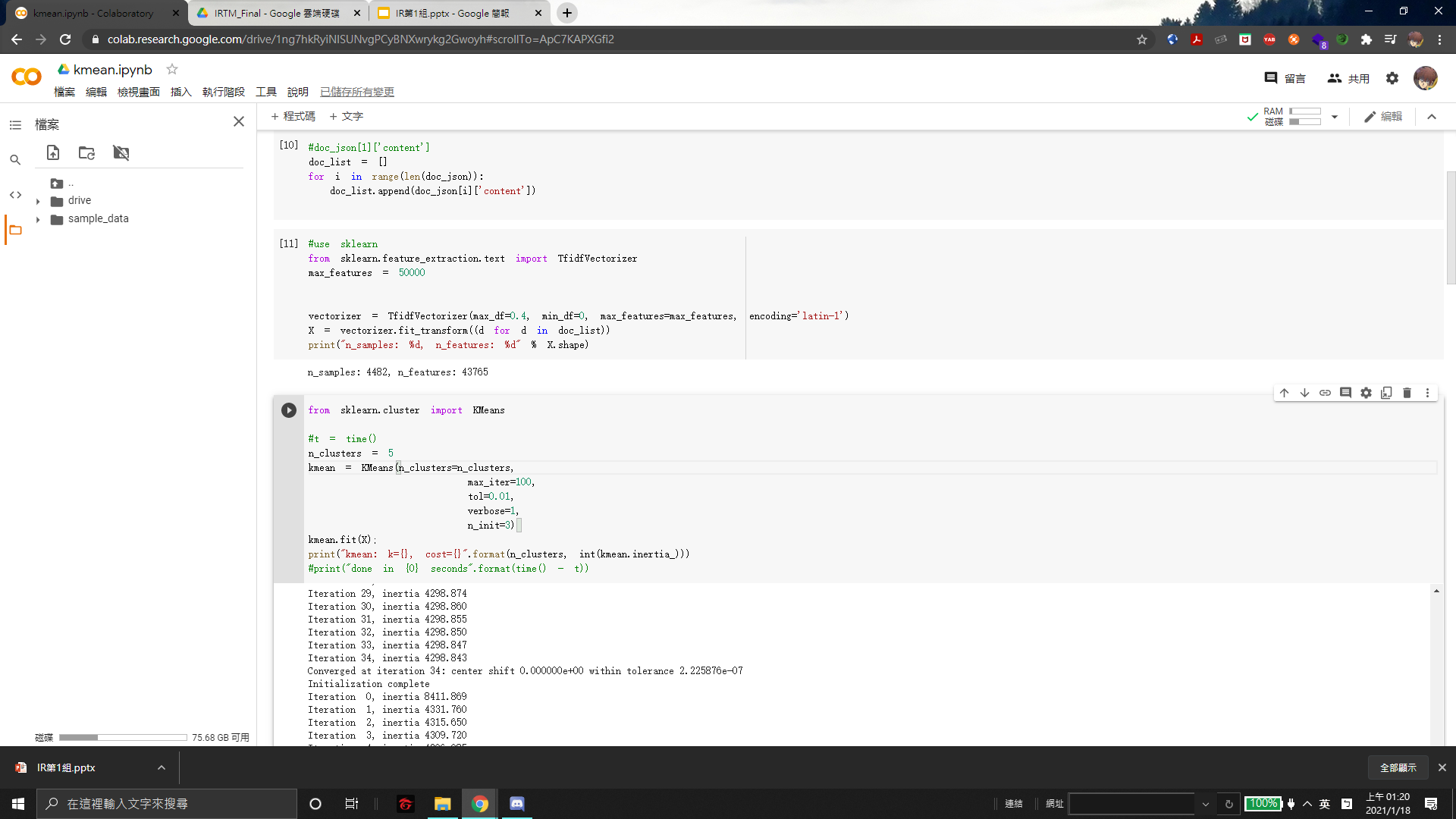
我們使用的 Model 是 GPT，然而效果不夠顯著，應該是因為我們的硬體資源不夠好(batch\_size太小)、還有訓練的時間不足所導致的。

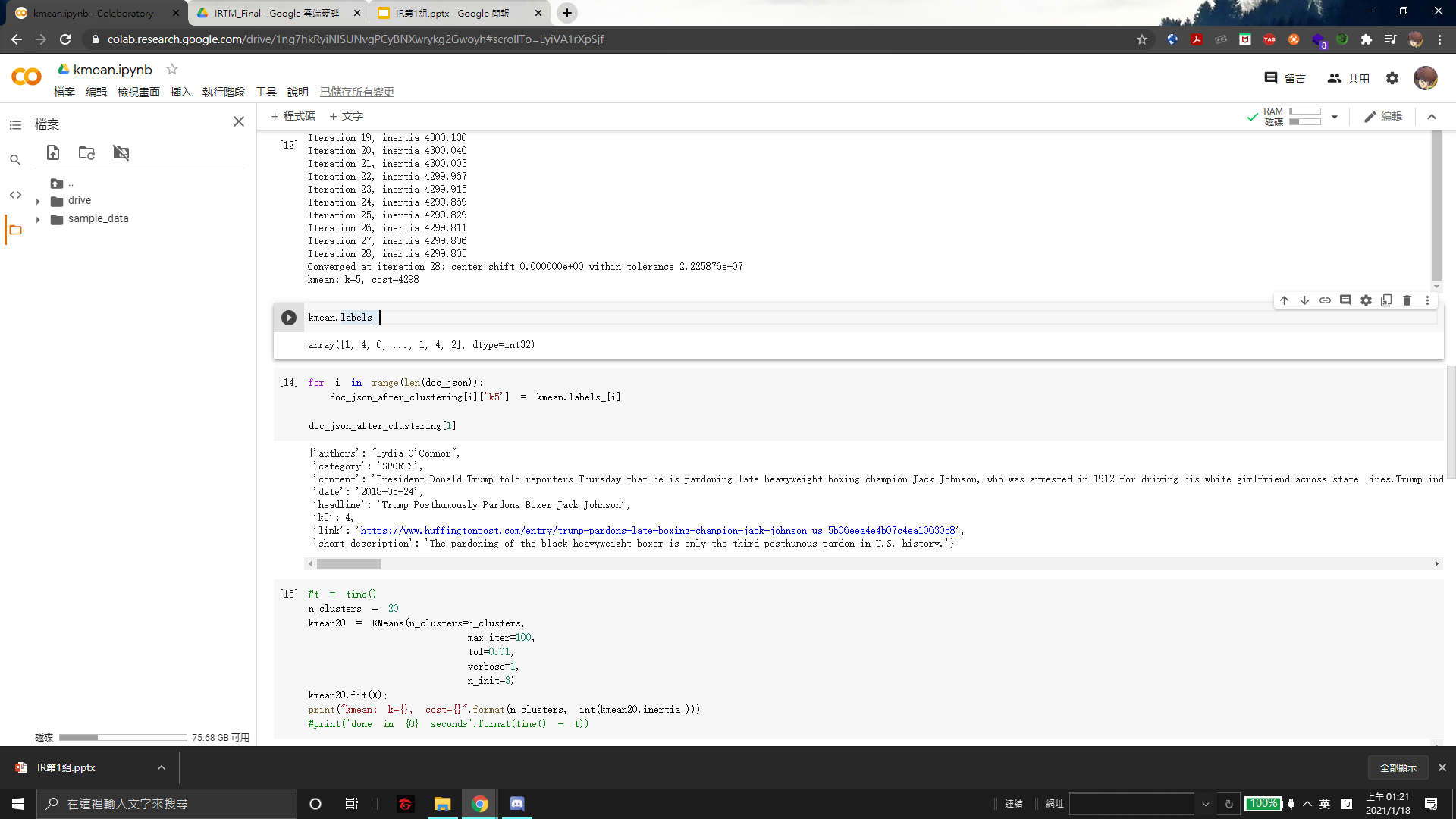


除了直接使用 Transformer，我們還有先將文章用 Kmeas 先初步分成 20 群，再小群小群的進行 training，可惜效果依然有限。

**Kmeans**

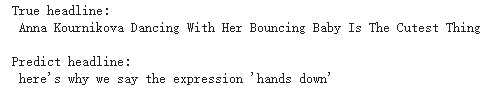






以上為Kmean分群的程式碼，使用的是sklearn套件，先用vectorizer將文件向量化。再宣告KMeans後以kmean.fit吃下向量，最後kmean.labels\_即可看到分群完後的結果，我們實行了5群及20群的兩種分群，並以分完群後的資料輔助Seq2Seq Model

**System Outcomes**

* Tansformer: 
* N-grams:

generated headline: The whole audience collapsed!!! Nets In a separate deal Houston is trading LeVert and Rodions Kurucs from the Cavaliers via the Milwaukee Bucks an!

**Conclusion**

在N-gram的模型中，以3-gram的結果最好，拼湊出的標題語意尚通順，同時也能擷取出文章的重點資訊，但無法透過擷取的資訊客製化出每篇新聞獨特的農場標題。模型改良方式可能會先從對新聞分類開始，將新聞分成更細的數個群，藉此驗證這些隨機產生的標題並進行篩選，得到更貼近文章內容的標題，

Deep learning的方法是採用seq2seq，由於我們對這個方法的瞭解還不夠深入，加上硬體設備不夠好以及資料太雜亂，造成結果出現overfitting的問題，效果反而比N-gram還要差，若我們對seq2seq有更多了解的話，應該可以得到比N-gram更好的結果才對。

**Workload**

* B06705002 黃啟宏: N-grams 實作、PPT 整理
* B06705050 李翊辰: N-grams 實作、上台報告
* B06705058 劉品枘: 資料集爬蟲、TF-IDF實作、Deep Learning 研究與實作、PPT 整理
* B06705021 許亦佑: Deep Learning 研究、Kmeans 實作、上台報告

**References**

<https://github.com/as-ideas/headliner>

<https://medium.com/%E7%A8%8B%E5%BC%8F%E5%B7%A5%E4%BD%9C%E7%B4%A1/autoencoder-%E4%B8%89-self-attention-transformer-c37f719d222>

<https://zhuanlan.zhihu.com/p/54578457>

<https://arxiv.org/pdf/1706.03762.pdf>