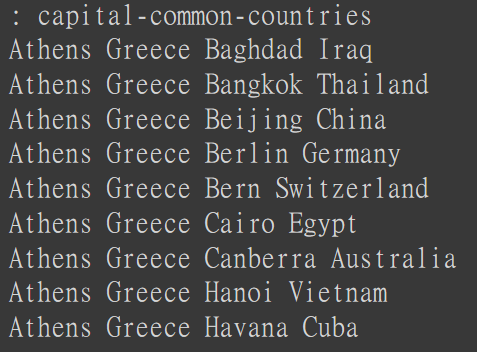
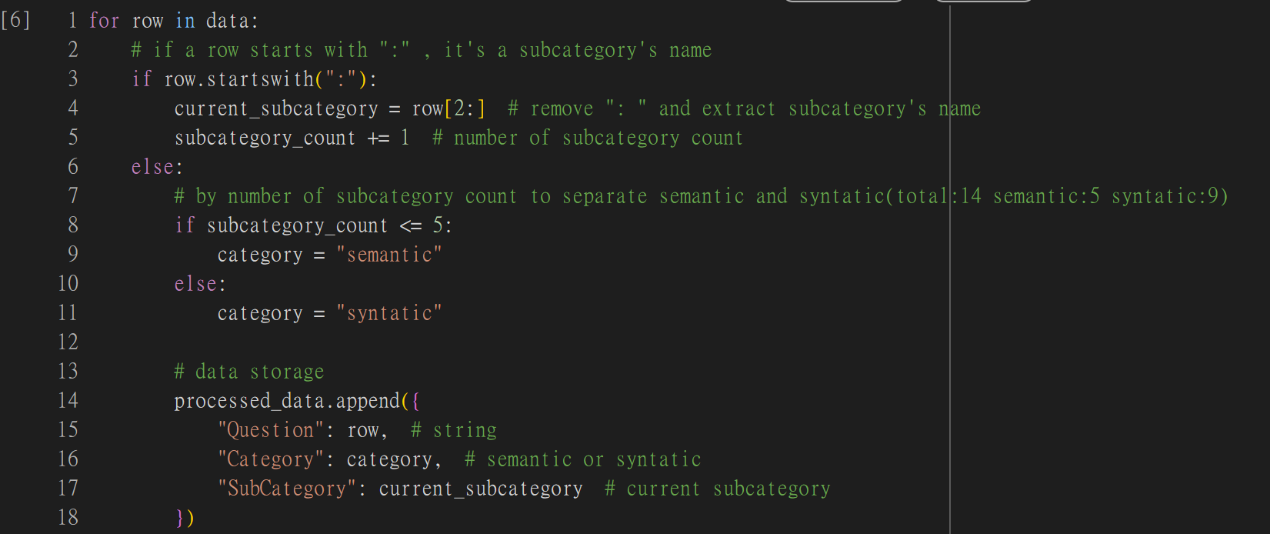
**2024 NTHU Natural Language Processing Assignment 1: Word Analogy NTU r12631055**

Running environment : Colab Python version : Colab

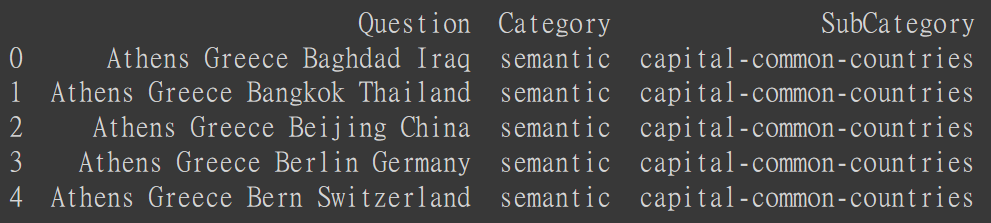
**TODO1**



根據raw data格式以及要求的輸出dataframe格式來設計程式:



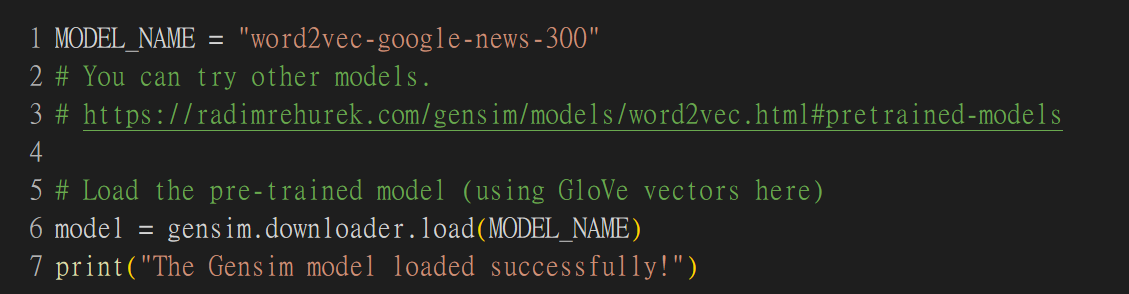
一行(row)一行載入data，因為格式都是以”: ”+”SubCategory”開頭，後續就是一連串的該子類別的字串，所以用”:”來作為開頭與分界來1.登錄SubCategory的字串、2.區分各字串的子類別類型以及3.作為計數來分類Category，據data info.前五個Category會是semantic，後續的則可能都是syntatic，做完條件式以後就一行一行append進入dataframe，最後結果如下：



後來發現好像不小心將Category內的分類”semantic””syntactic”弄成小寫開頭，”syntactic”錯拼成”syntatic”，但似乎不影響後續處理:D。

**TODO2**

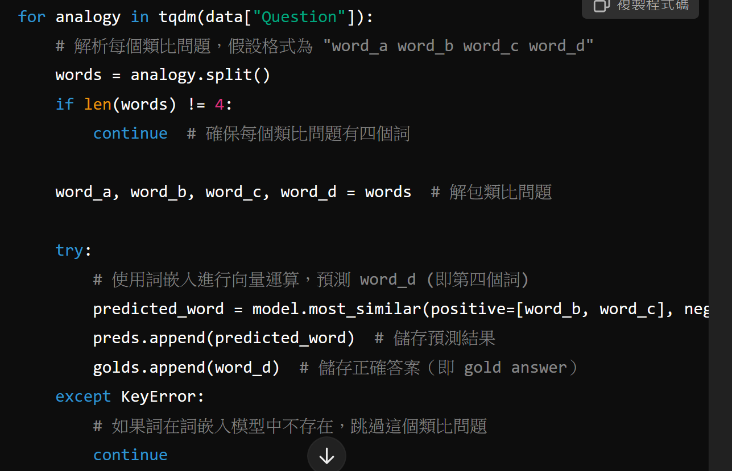
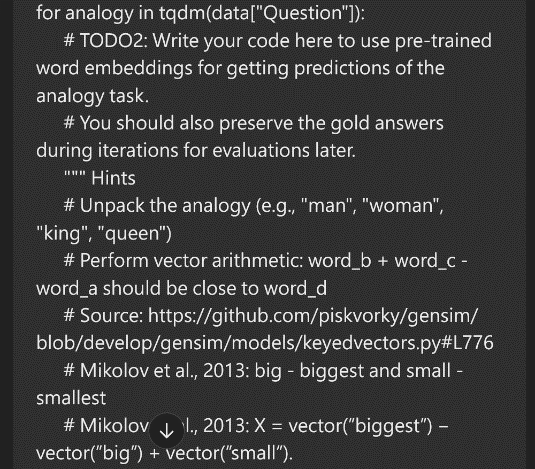
關於pre-trained model，實際測試過'glove-wiki-gigaword-50', 'glove-wiki-gigaword-100', 'glove-wiki-gigaword-200', 'glove-twitter-25', 'glove-twitter-50', 'glove-twitter-100', 'glove-twitter-200',等模型發現其字庫都沒有完整涵蓋整個dataset的字詞，特別是在地名資料上的缺失最多，因此選'word2vec-google-news-300'其涵蓋最為完整。

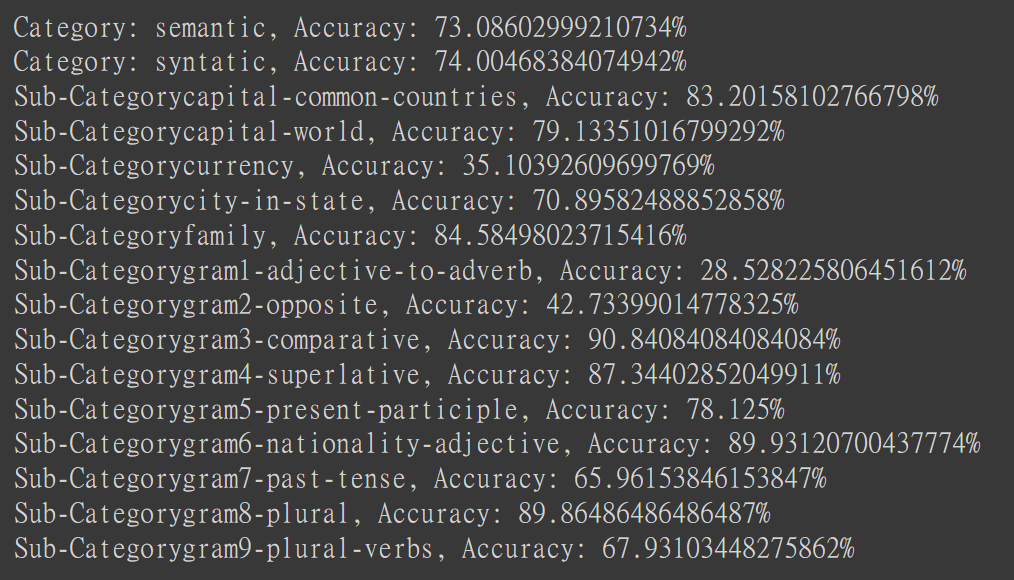


然而向量長度是300，因此做analogy也特別久(Colab2小時18分)

因為沒有實做過使用gensim套件預測，所以詢問了chatGPT：

使用預訓練詞嵌入模型來解決類比問題，需要從每個類比問題中提取四個詞，根據詞嵌入向量運算（word\_b + word\_c - word\_a）來預測第四個詞。

最終預測結果如下:



Category方面：”Sematic”與”Syntatic”的Accuracy其實差不多，說明整體上pre-trained model在語意和語法分析上的表現都不錯，但我們應該需要去細看子類別區分才能得到更好的推測。

從子類別來看，”Sematic”中表現最好的是family與capital-common-coutries，而表現最差的是currency，至於”syntatic”中表現最好的是comparative、nationality-adjective、plural與superlative，而表現最差的則是adjective-to-adverb和opposite，遠低於其他子類別。

對於這結果我的猜測：首先是semantic的部分，因為'word2vec-google-news-300'的文本可能是來自於各式各樣的新聞文本，因此在語意方面，在國家與首都地名上能得益於世界新聞報導(英語世界經常以首都地名代指該國政府)，以及在親族關係上能夠得益於各類社會花邊新聞:D。

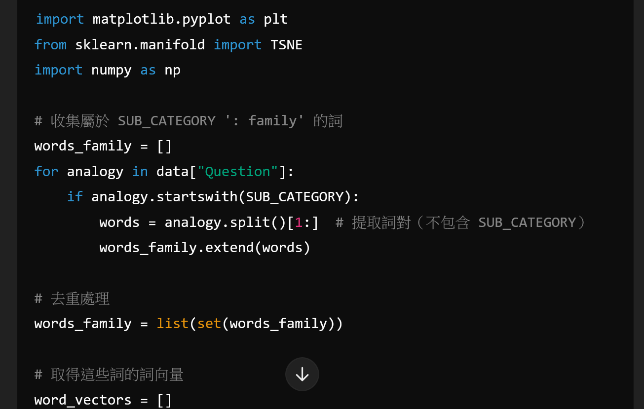
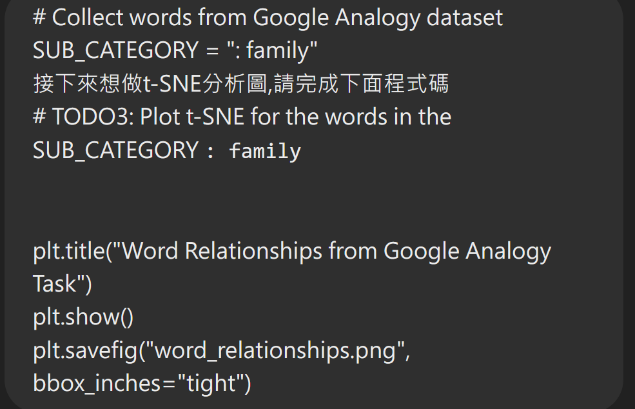
而至於在貨幣方面雖然有財經匯率相關的新聞，但整體上財經新聞報導基本上只會著重在主要貨幣上，如美元日圓等，較少有針對於次要貨幣則報導，比例相當懸殊，因此預訓練模型在這方面可能會表現得不好。

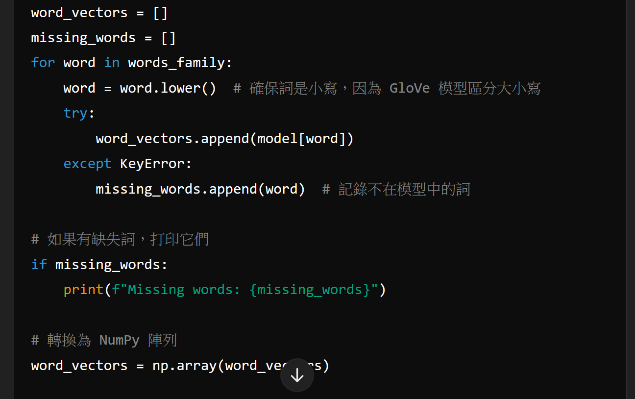
而觀察syntactic的部分，其實從整體上來看表現得還算不錯，其中只有形容詞到副詞的轉換和相反詞表現特別差。個人猜測有可能是新聞文本多半都是以陳述人物事件現象為主，對於從形容詞到副詞的轉換和相反詞呈現等等在文章中並不會有太多機會使用到，也因此可以想見模型會在名詞單複數上以及在國家與國家形容詞上這類人事物陳述的表現特別良好。

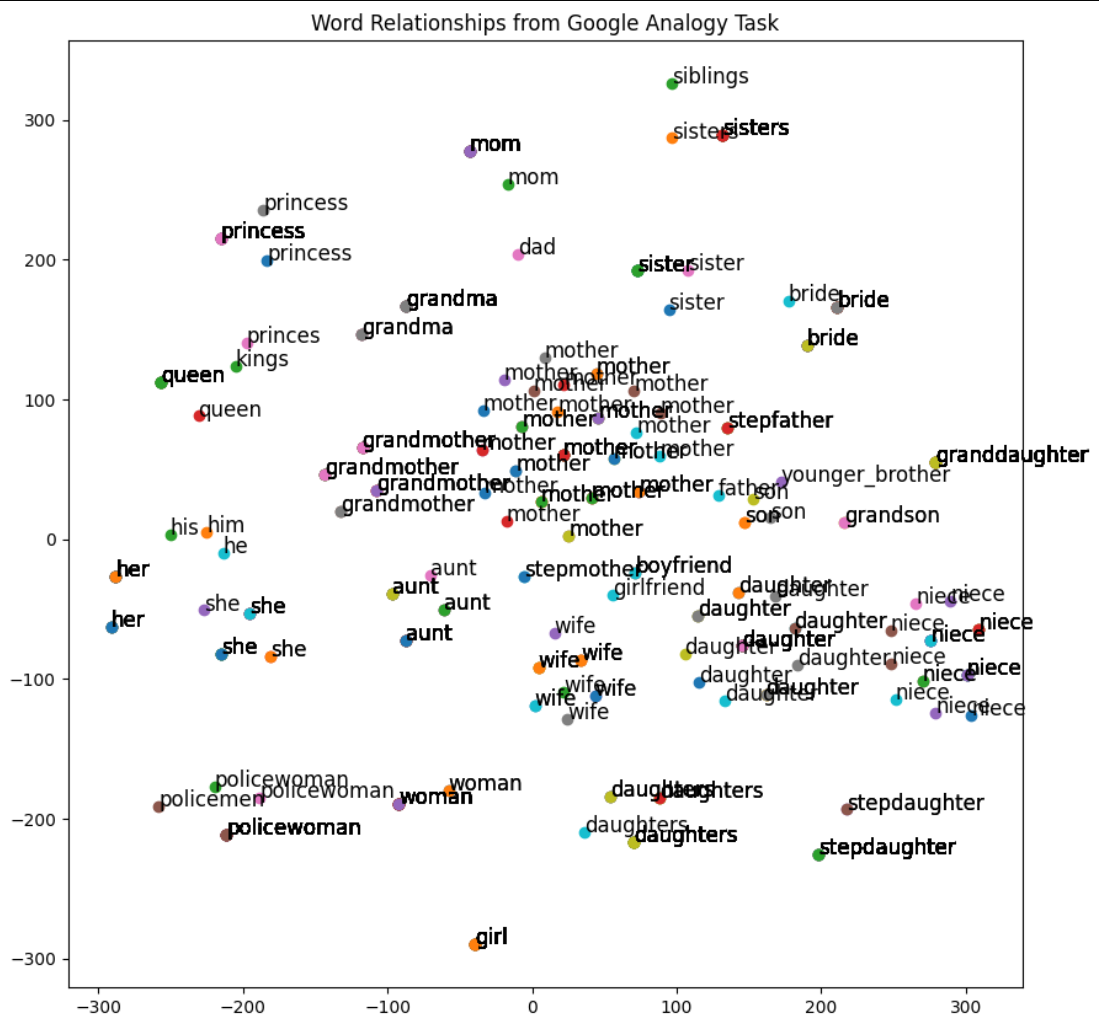
我們同時也能猜測，在新聞文本的文章陳述當中，比較喜歡同時使用同一字詞的比較級與最高級來進行陳述，而比較少會做形容詞到副詞的轉換與相反詞的呈現。這也是除了前面所述的新聞文本以陳述人物事件現象為主以外的，可觀察到的新聞文章陳述語式偏好。

**TODO3**

第一次接觸到t-SNE繪圖，因此詢問了chatGPT該如何進行：







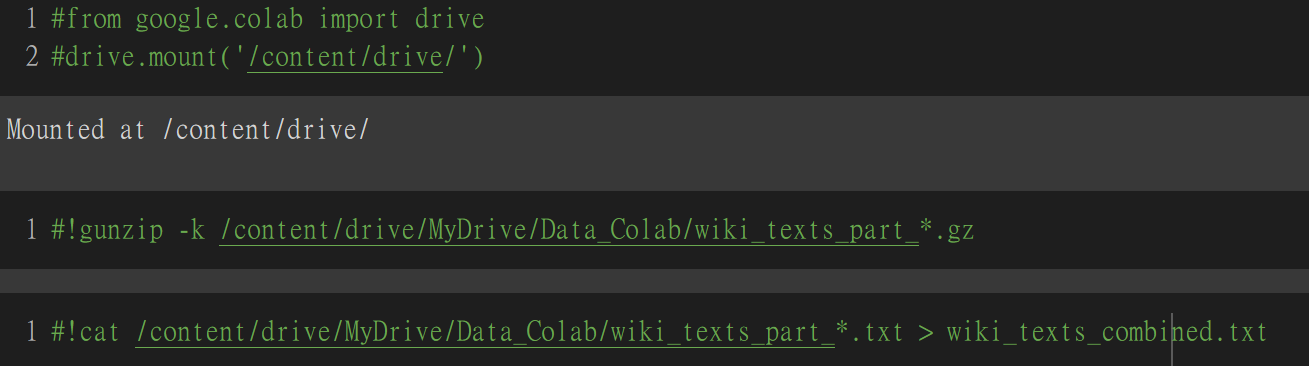
上面這張圖是在做deduplication前的t-SNE，可以清楚看到相同語意的詞的位置都非常接近，這表示model是確實有抓到這些詞之間的關係的。而下一張則是做完deduplication(隨機取一)的t-SNE：



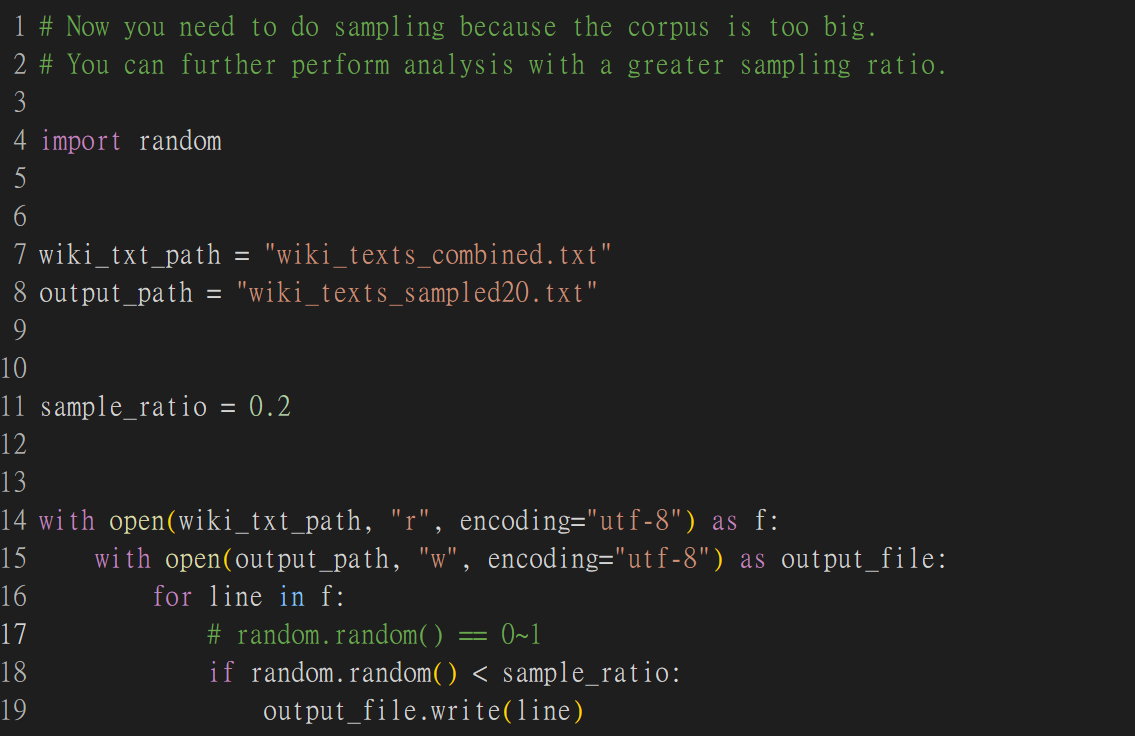
可以看到詞意相近或者關聯性高的詞的位置都相當接近。左下角是一整個家族相關詞的區塊，可以看到手足跟姊妹弟弟很接近，孫子孫女、外甥阿姨、繼母繼女以及父母兒子女兒都出現在相當接近的位置。而外圍的一大圈則是呈現兩兩接近的狀態，男警女警、第三人稱詞、男友女友等，而公主皇后同時也跟王子國王接近。

**TODO4**

接下來4跟5花費了比較多的時間，原因是處裡很多相當大的資料，而Colab分配的記憶體又相當有限，常常發生有些部分因為記憶體耗盡導致整個作業重來的情況，因此我自己在處理時多半直接將檔案mount進gdrive內，避免crash後還要再整個重新跑一遍：



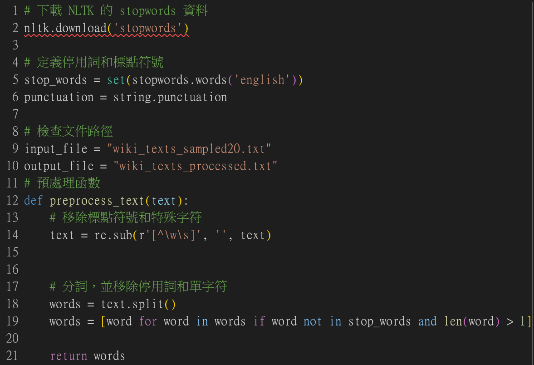
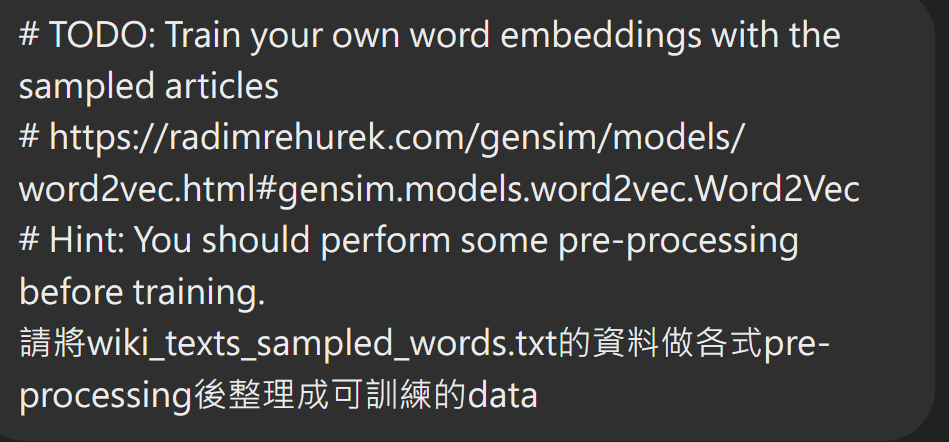
而在sampling 20%這個地方就是經常跑到RAM耗盡的其中之一，我個人有嘗試過很多不同種的取法：例如每一行都隨機取20%的word(但發現會破壞上下文關聯性)、或取一整個chunk的20%句子



而最後還是因為記憶體的關係，所以選擇使用每一行(Row)都有20%機會被取樣，並立刻寫入txt避免浪費RAM的方法，如上圖所示。

**TODO5**

因為第一次做text的preprocessing，所以還是先詢問了chatGPT：

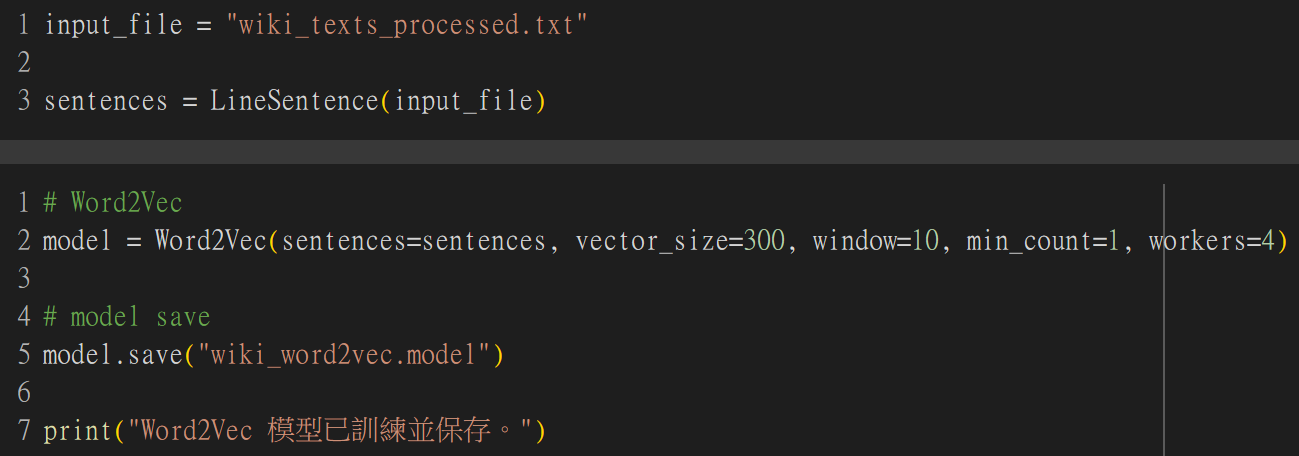


也如同前面取樣一行一行作preprocessing，其中包含了：

1. NLTK 的 stopwords (english)
2. RE去除標點符號和特殊字符
3. 利用text間的空格作為分詞標記來進行分詞

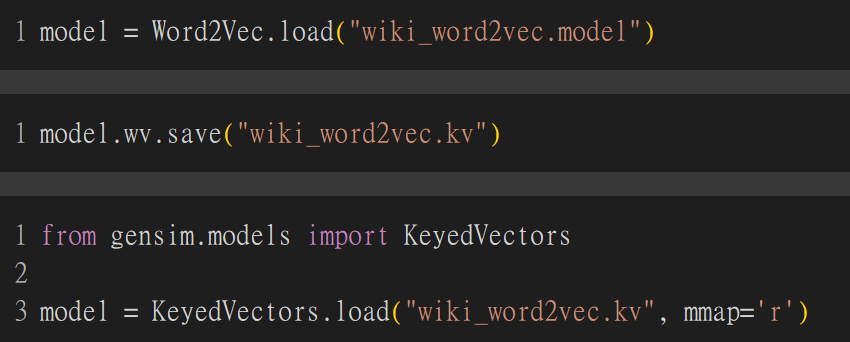
至於Lemmatization的部分，考量到Google Word Analogy其中有項plural的subcategory要做單複數關聯，怕缺失因此就不進行處理。

考量到想盡可能避免有詞缺失，因此我嘗試把vector size設為300， window = 10，min count = 1；為此光做一次訓練就要花費大約6小時而RAM也常會耗盡，所以得用LineSentence來餵進Word2Vec：

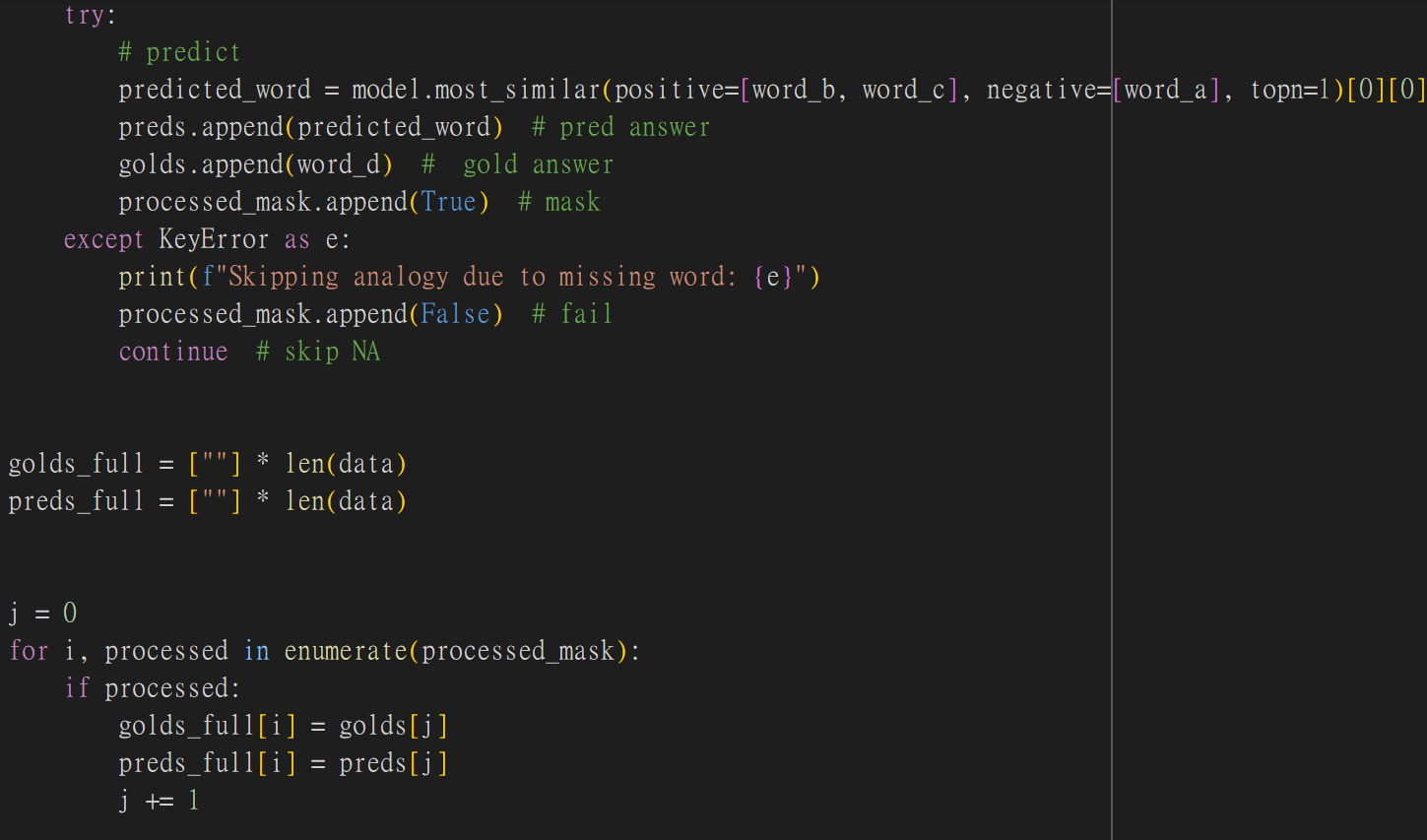


**TODO6**

接著參考TODO2的model部分修改程式碼，因為光是model載入就已經消耗太多記憶體了，所以將model轉存成KeyedVectors(詞向量)來代替完整模型，這樣一來RAM的消耗便可以變小許多。

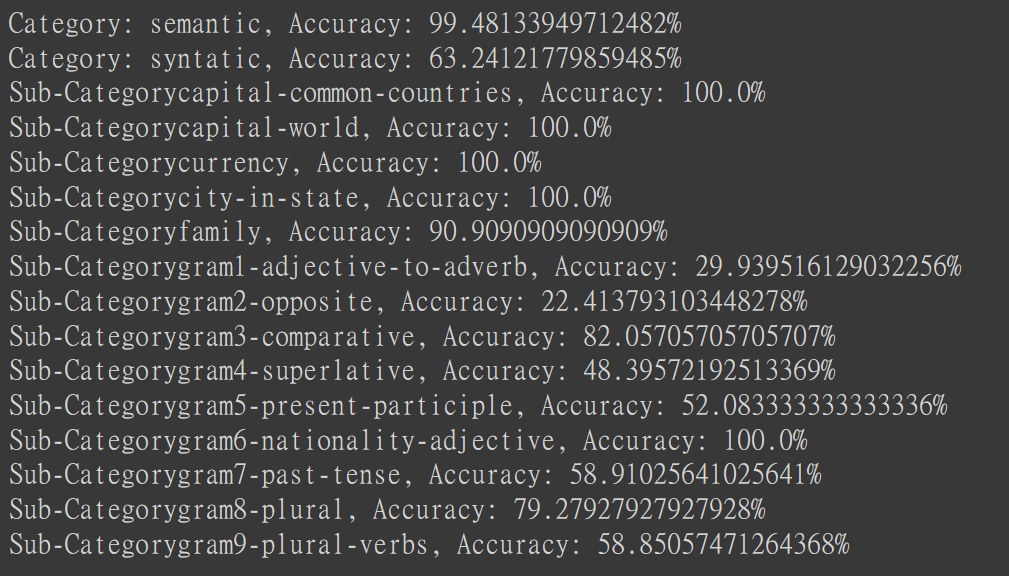


Predict的部分則因為model裡面還是有很多詞有缺失，所以需要再進行一些修改以及跟正確答案(gold answer)格式的對齊，但是因為model很大，最後完成後每進行一次predict大約就快要花2小時。

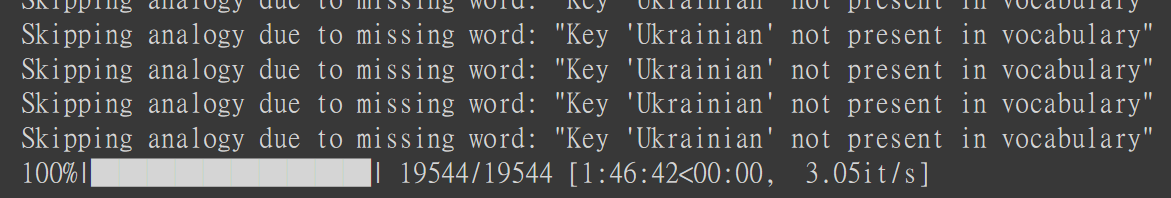


**TODO7**

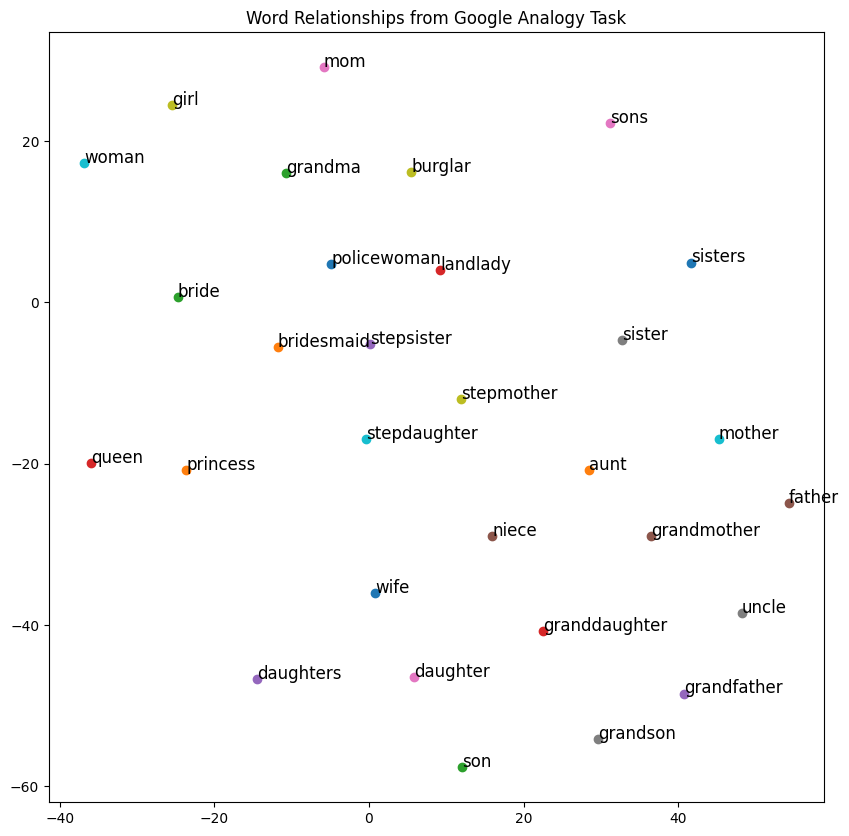
最後是預測結果與t-SNE的環節，首先是Accuracy的部分：



乍看之下似乎在semantic的正確率很高，然而實際上是因為很多辭彙都缺失而無法計入評分當中，在19544項類比問題中，實際上沒有詞彙缺失(亦即能進行評分的)的項目大約不足一萬，而詞彙缺失這方面又以如國家、地名、第三人稱詞彙(會被stopwords去掉)最多，相較之下syntactic的表現感覺才比較能夠反映預測真實情況，而部分高正確率雖然難以衡量真實狀況，但也許能猜測在pre-trained model有詞彙的情況下，預測能力還可以算是準確，可能是高vector size與高window值的影響，若進一步增加資料量也許還可以更好：



最後則是在t-SNE的部分沿用TODO3的程式碼，其結果如下：



直感上整體表現雖然並沒有像是在TODO3時'word2vec-google-news-300'的pre-trained model那麼好，但還是可以看出有一些分布的趨勢存在，如相關的家族稱謂之間整體還是比較靠近，而像女兒單數複數、公主女王、新娘伴娘以及女人女孩等也是有兩兩相近的趨勢存在。Training data數量不只對整體趨勢有影響，也會影響到詞彙缺失的狀況：經過wiki data實際測試在較少的Training data(5%,10%)的狀況下，詞彙缺失的情況更嚴重，而且整體類比預測的準確率更低。