政大商院有嘻哈

之使用RNN生成嘻哈風格歌詞

105703003賴弘易 105703007陳靖元 105703015謝忠凱 105703052張淯淞

（一）前言:

當初知道期末要準備project，便開始找尋有關深度學習的應用，像是深度學習可以利用大量數據分析的方式，來預測事件結果。但這些對我們來說需要花較多時間，於是找到一個比較簡單 ，我們也比較容易做得出來的應用---生成嘻哈風格歌詞。之所以會找這個主題除了是我們做得出來外，會用到現在很流行的網路爬蟲、NN(神經網絡)model等方法，希望也能在這個project中多多磨練。

（二）步驟:

2.處理爬蟲下來的資料，將不需要的資料刪除

1.網路爬蟲

從KKBOX抓歌詞

3.使用jieba套件分詞

4.把歌詞轉成Tensor儲存

6.得到training結果，將Loss值視覺化，並嘗試

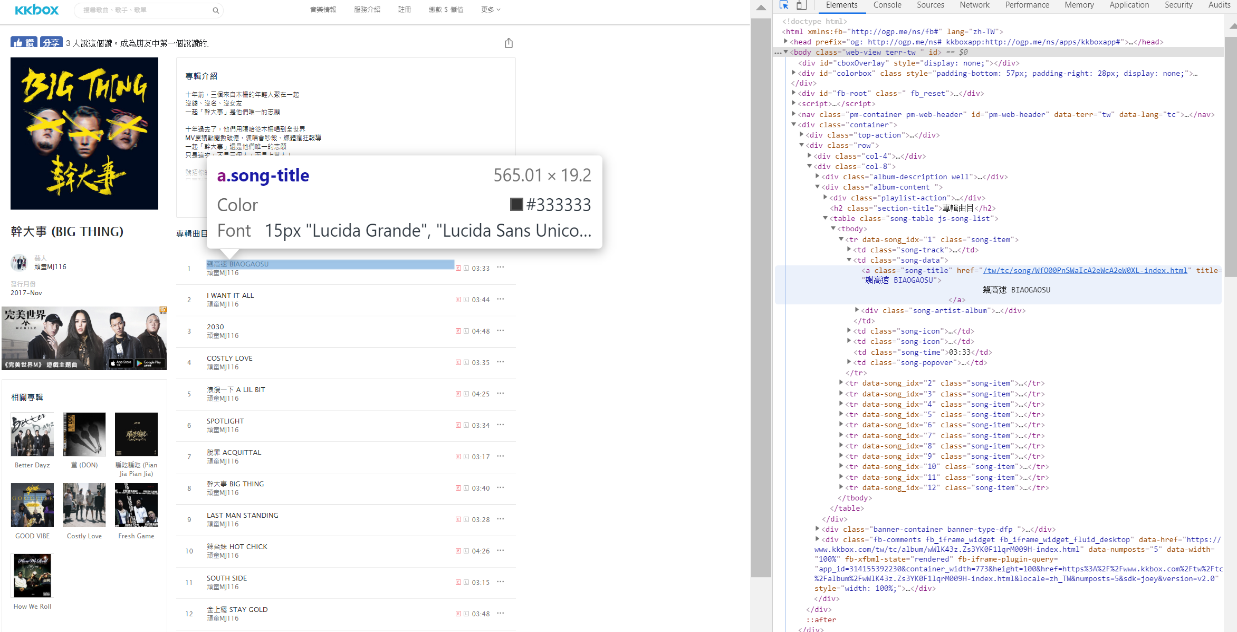
修改資料與模型

5.放入RNN model 進行training

（三）實作方法:

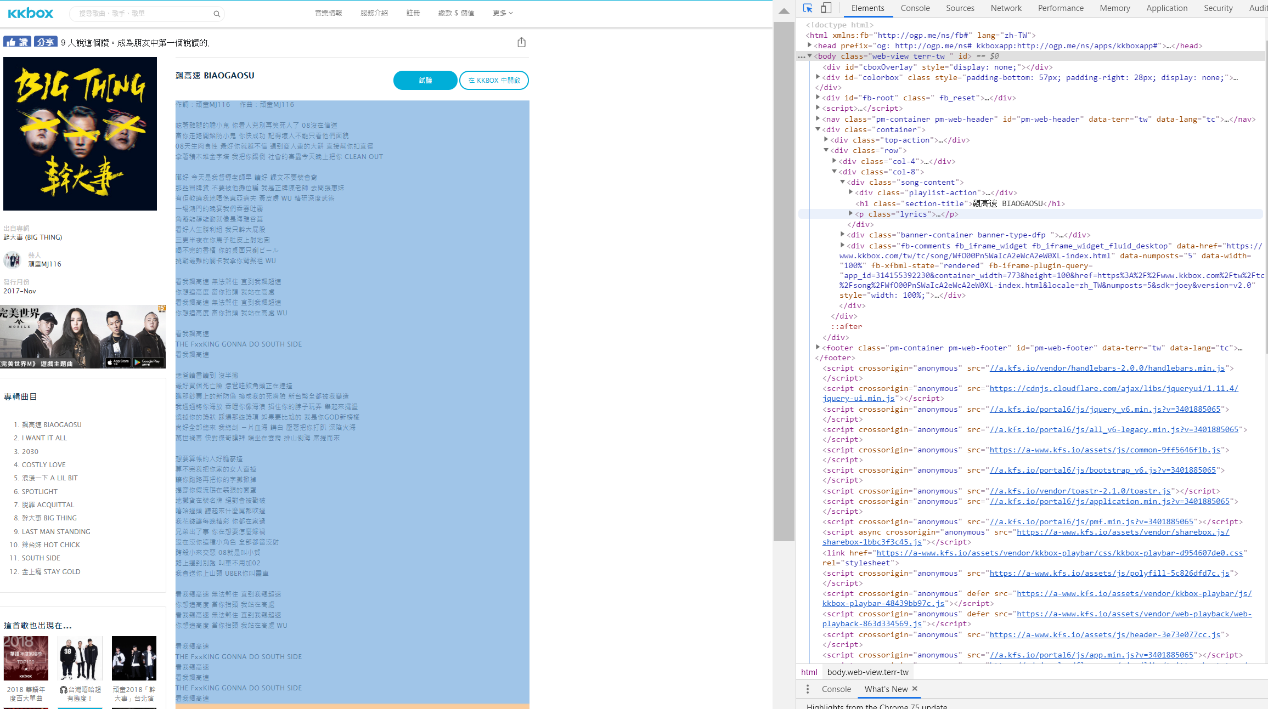
1. 用網路爬蟲爬取歌詞

這裡的爬取歌詞方式使用到了一套Python的Framework-Scrapy，可以迅速建立爬蟲程式，我們先爬取知名音樂串流平台KKBOX網站中台灣各嘻哈歌手專輯頁面中所有曲目的連結，將所有我們想要爬取的歌建成文字檔，再使用第二支爬蟲程式去讀取所有歌曲的連結，分別到其頁面將每首歌的歌詞爬出，再建立成文字檔。



爬取該專輯有歌曲之連結

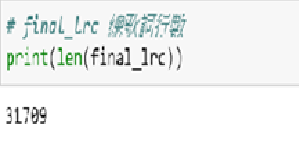


每首歌都有顯示歌詞的頁面

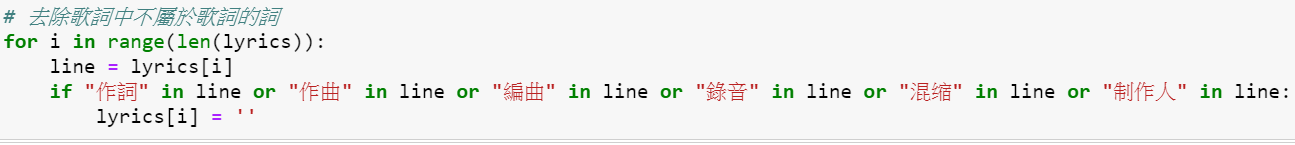
將該頁面歌詞爬取的程式碼

1. 處理歌詞成input資料

我們將從KKBOX 爬取了31709行的嘻哈歌詞



接著我們要從這些無數的歌詞中刪除一些不需要用的信息，像是作詞」，「作曲」等信息，這對訓練模型生成歌詞是沒有任何幫助的，先刪除他們。

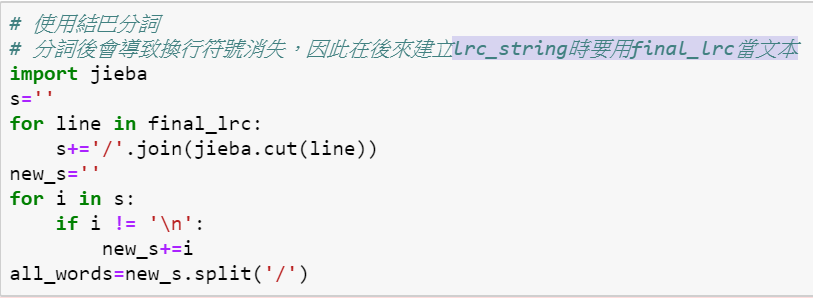


再來，因為嘻哈的特色，所以很多歌詞都會有英文歌詞，但訓練數據中存在的語言種類越多，意味著深度神經網絡模型學習起來的難度越大。所以我們打算把英文歌詞去除掉，不打算讓模型去預測英文的部分



然後，我們使用結巴分詞 ，但分詞後會導致換行符號消失，因此在後來

建立lrc\_string時要用final\_lrc當文本

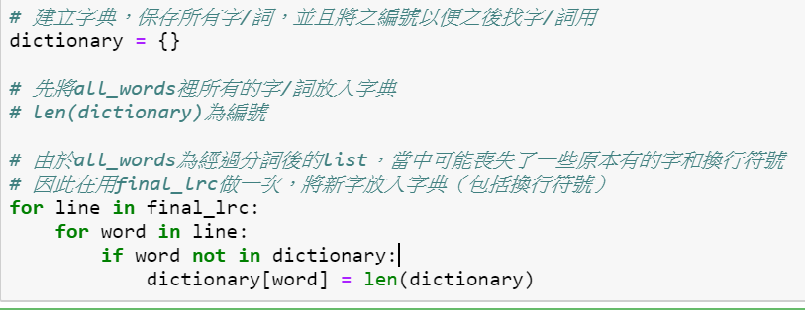


之後建立一個字典，把所有的字丟入字典裡，並且把它們作編號以方便

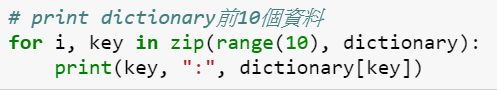
之後找字，由於all\_words為經過分詞後的list，當中可能喪失了一些

原本有的字和換行符號，因此在用final\_lrc做一次，將新字放入字典

，包括換行符號。

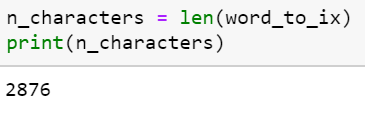


列印出字典中的前10個元素，注意它是無序的



除了建立dictionary 外，還需要建立一個「有序列表」，有序列表就是將「字典」按照索引號從小到大的順序排序。

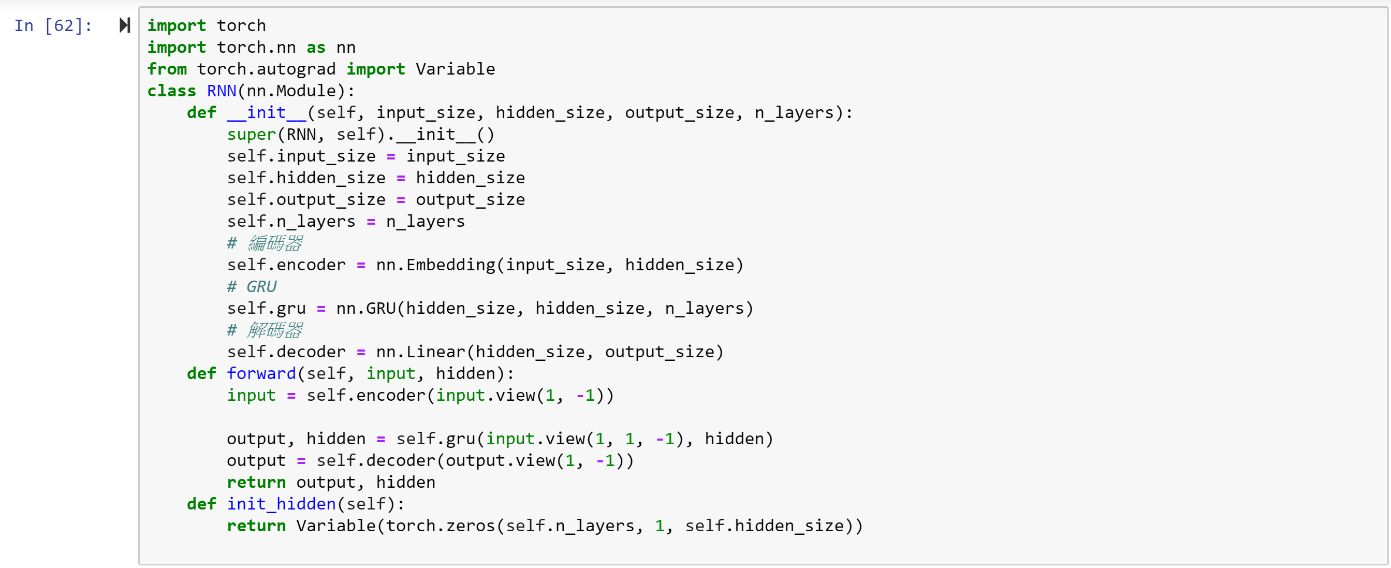
「字典」的長度代表訓練數據中有多少不同的字，這個長度即是我們模型輸入層的大小，我們建立一個變量 n\_characters 來保存它。



(3)使用RNN model生成歌詞

考慮到生成歌詞所涉及到的為自然語言的處理的領域，我們第一個想到的就是這個領域最常使用的RNN模型，因此選擇RNN模型來traing我們的歌詞。

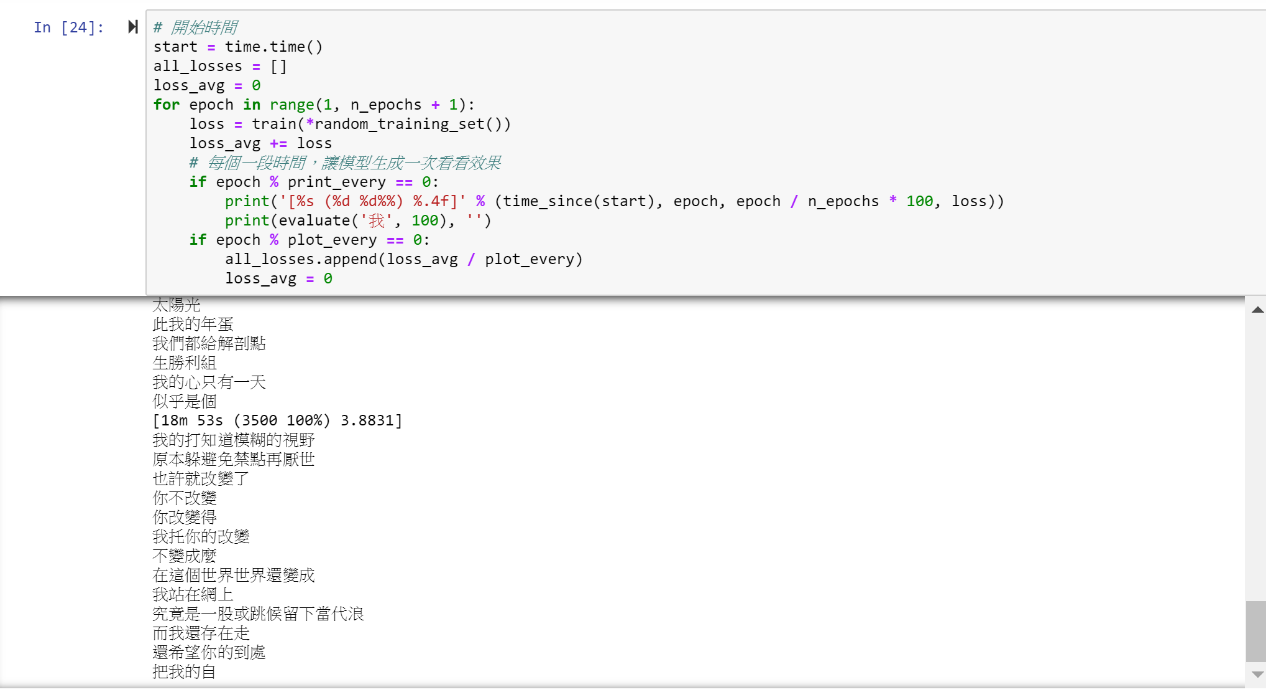
Training Model



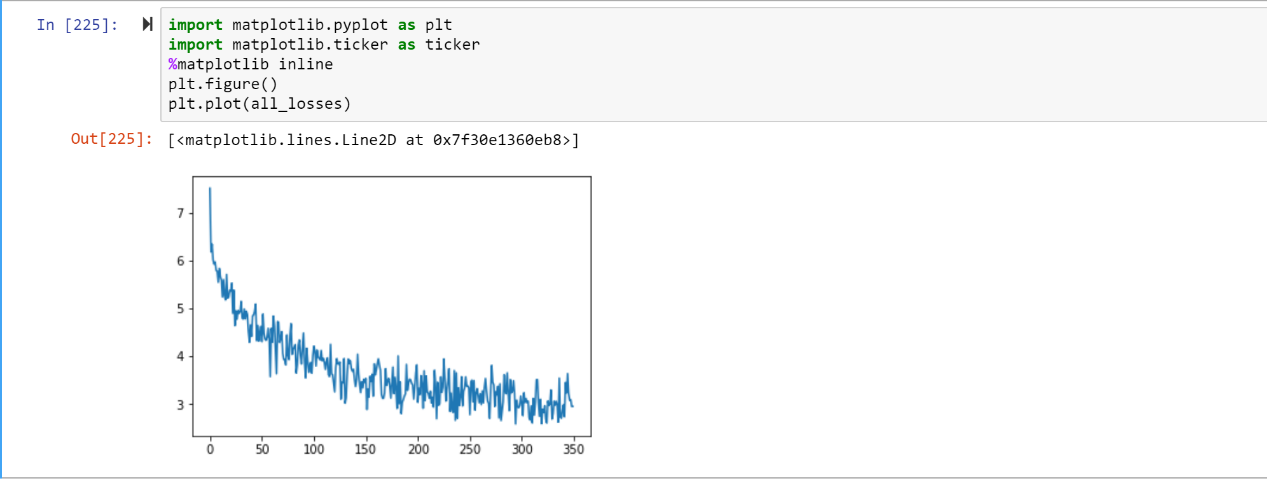
第一次訓練我們將爬取的一萬多行資料先放入model訓練，epoch設為3500次、隱藏單元的數量設為256個，隱藏層的層數設為兩層，學習率設為0.001，隨機採樣部分歌詞選擇200字，並將RNN model設為decoder，使用Adam優化算法，用交叉熵損失作為Criterion。

Training的結果產生出的歌詞有些部分還不能符合正常的語意，因此我們決定繼續嘗試修改參數與增加放入model歌詞量，希望得到更低的loss值，看看training出的歌詞會不會更好。

第一次training結果

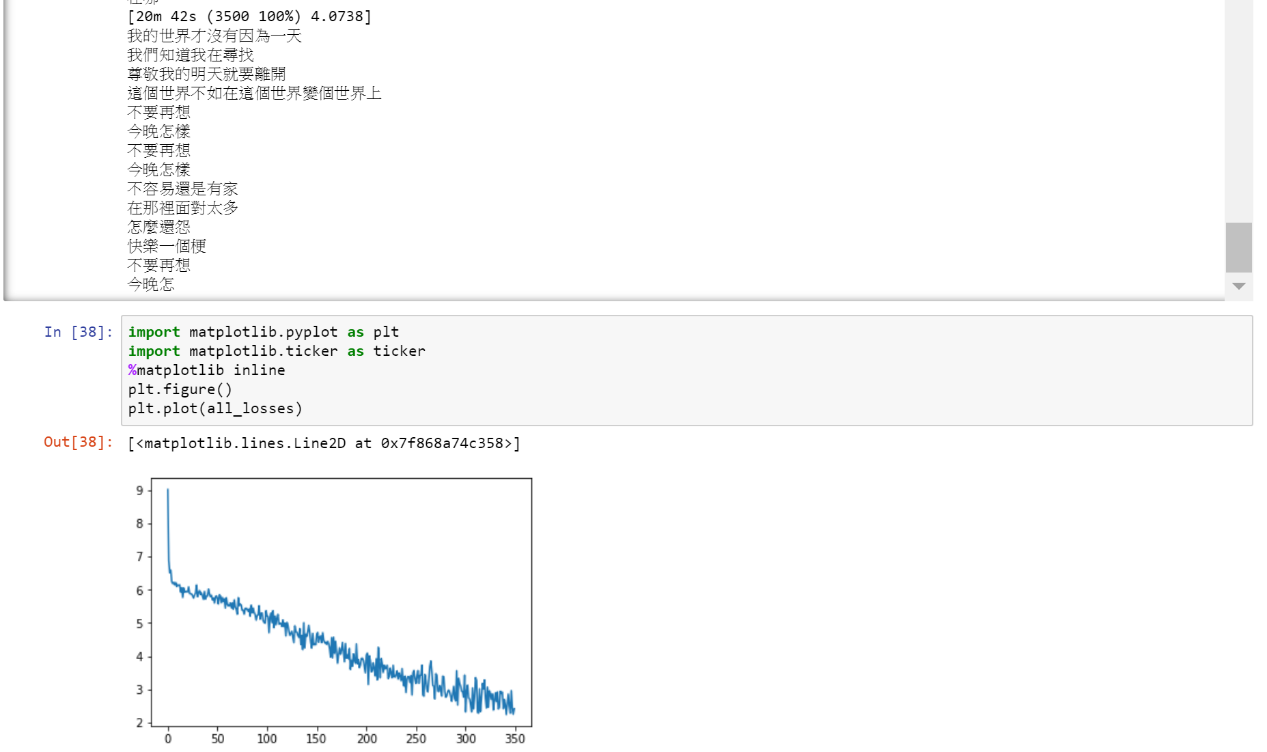


Loss值並沒有收斂



第二次Traing前，我們在其他同學的建議下，決定將歌詞進行分詞，增加training data的複雜度，並且將歌詞量從一萬多行增加到三萬多行，但結果也還是沒有生成語意相較於上一次更多通順的歌詞，loss值最低也還是在2到3上下，之後我們又將隨機採樣部分歌詞量選擇增加到10000字，且把學習率從0.001降到0.0001，但這些改變對於loss都沒有顯著的影響。

第二次training 結果



(四)反思與改進:

經過多次資料處理以及調整不同參數的training後，我們得到的結論是讓資料量增加，會讓loss值些許上升，但增加分詞以及對learning rate、epoch、隱藏單元的數量、隱藏層的層數調教，對於loss值的影響實在不大，

我們認為可能的原因為我們沒有將Training Model完全建好，或是RNN本身的侷限，針對這點我們之後可能會再繼續嘗試將RNN 改成Seq2Seq、GAN以及Skip-gram 這幾種model，與RNN比較看看結果是否更好；此外，訓練的句子沒有處理好我們也認為是造成訓練結果不夠好的原因，這點目前想到的處理方式是Embedding的部分使用pre-trained model。透過這個Project的實作，我們都對深度學習有更進一步的了解，希望我們都能將這堂課所學到的深度學習研究方法與知識運用到日後我們所接觸的領域上。

(五)分工表：

程式碼：賴弘易、張淯淞、陳靖元

PPT報告：賴弘易

Word文書：張淯淞、謝忠凱、陳靖元

資料查找：全體