你有 Freestyle 嗎? 中文饒舌歌詞產生器

林芳瑀 R05725009
郭哲偉 R05725010
蔡曜宇 R05725012
吳洺甄 R05725030
彭毅軒 R06725038

1. PURPOSE

近年來，人工智慧被應用在「創作」並成功的例子劇增，Taryn Southern 新專輯的首支單曲《Break Free》在 2017 年 8 月 21 日發表，其中的編曲創作即百分之百來自於人工智慧。雖然人工智慧尚無法完全取代人類進行感性的創作，但其異於人的思維模式及產出之速度，有益於幫助人以更有效率的方式完成創作。同年，中國綜藝節目「中國有嘻哈」引起了「你有 Freestyle 嗎？」的熱潮，成功為嘻哈音樂帶來活絡新血，也讓更多音樂愛好者能享受嘻哈文化。其中，歌詞的創作是決定一首饒舌歌曲品質優劣的關鍵，而好的饒舌歌詞不只注重於韻腳，歌詞能充滿生活省思、意義及詩意才能打動人心。前面所提到的人工智慧產生歌詞為英文歌曲，這也促發我們對於如何將機器學習方法運用產生中文嘻哈歌詞的學習動機。

然而在找尋各大歌曲與歌詞網站時，我們發現目前各大音樂網站在中文嘻哈歌曲的細部分類都是以說唱方式或是地區類型來進行分類，如此一來，在根據不同場合與情況中，像是派對、失戀療傷和抒發對生活不滿等等不同的需求情況下，單純以說唱方式或地區來進行分類的結果不一定能找到符合當時情況的嘻哈歌曲內容，並且也無從瞭解這類歌詞比較傾向甚麼類型的嘻哈歌曲內容來進行嘻哈歌曲推薦。

所以此專案的目的為，運用課堂所學的機器學習模型對大量的中文嘻哈歌曲進行有效的歌曲分群分析，並且將每一群的特徵描繪出來，以此瞭解目前中文嘻哈歌曲的風格內容為何，但是我們對於中國嘻哈歌曲的內容並不未全面的認識，所以我們將嘗試運用課堂上所教的分群方法來自動進行分群並找尋主題以協助我們更瞭解嘻哈歌曲的類型與內容。最後我們運用 Long Short-Term Memory language model (LSTM) 來自動產生符合作詞人個人特色之歌詞，過去此類型之模型必須藉由外力定義歌詞生成之模板，而 LSTM 可以做到自動定義每句歌詞之長度、韻腳，結果顯示其產生之歌詞優於其他 Baseline model。我們在資料集的選用上著重於來自於中國、台灣及香港等亞洲地區的嘻哈創作歌詞，利用 LSTM 學習創作者的編詞模式，期待能提供嘻哈音樂愛好者與音樂人更多的創作靈感。

2. SOLUSION

根據上述所提及的動機與目的，我們希望能夠針對嘻哈說唱歌詞進一步分析並將課堂中所學習的機器學習知識進行模型建立來幫助嘻哈歌手或是一般聽眾來快速瞭解此歌曲的風格內容甚至更進一步創造出具有該風格特色的饒舌歌詞。主要是從蝦米音樂網站爬取嘻哈歌詞，經由歌詞斷字與前處理後作為訓練資料，接著運用此訓練資料進行分群模型建立與歌詞產生器模型以提供機器學習方法來解決目前所遇到的困難點，並且有效幫助嘻哈歌詞的創作與瞭解。我們主要以 Python 語言來進行程式實作，運用文字探勘與資訊檢索的技術還有深度學習演算法來建立這些功能，以下我們將更詳述整體實作流程與技術內容。

2.1 資料蒐集

我們選以蝦米音樂 (http://www.xiami.com/) 作為歌詞蒐集來源，蝦米音樂的音樂分類較其他音樂網站齊全，從音樂風格上就分做 23 類，其中我們選以「嘻哈 (說唱) Hip Hop」類作歌詞收集 (http://www.xiami.com/genre/detail/gid/1)，以嘻哈代表藝人 (http://www.xiami.com/genre/artists/gid/1) 為單位 (共有 5760 位)，並只收集來自「China 中國」及「Taiwan 台灣」兩地區藝人 (共 3350 位) 的前 20 首熱門嘻哈單曲 (例如：頑童 MJ116 的前 20 首熱門單曲 [http://i.xiami.com/mj116/top])。但其中，有些歌曲只是純音樂，而無歌詞成分，故再經過篩選之後，保留具有歌詞內容的音樂，總共收集到 11095 首嘻哈說唱單曲的歌詞，作為我們的研究資料集。

2.2 結巴斷詞與前處理

由於我們想分析的主要目標是中文與英文歌，因此我們使用 Python 的套件 － Jieba 結巴中文分詞程式進行斷字，Jieba 除了可以在一連串的中文字中找出各個有意義的詞

語，也會用特殊標點符號進行斷句，將每首歌統計出各個詞彙的次數，以便後續分析。

接著，為了後續敘述統計、分群以及歌詞產生，我們需要將適當的移除歌詞中不必要的字元 (特殊標點符號)。我們將大寫英文單字的轉為小寫、全形轉半形，再使用 NLTK (Natural Language Toolkit) 的 Lemmatizer 做單字的 Normalization。以及歌詞中常常出現其他國家的文字 (韓文、日文)，我們也一併將之刪除。但是在 Stopwords 方面則要視不同分析方法再決定要不要去除。

最後，將爬下來的 11095 首歌詞經由歌詞斷詞與前處理後，將歌手相關資訊，例如：歌曲名字、歌手名字和作曲家等資訊以列記錄為一張資料表，接著再將每首經由斷詞和前處理好的歌詞內容以列的方式儲存為兩張資料表，其中分別是有去除 Stopwords 和沒有去除 Stopwords，將有去除 Stopwords 的資料表運用於後續的分群模型建立與統計分析，沒有去除 Stopwords 的資料表運用於歌詞產生模型建立。

2.3 敘述統計分析

在此敘述統計分析階段中，我們會先將 Stopwords 去除，剩下較有主題性的字詞，讓我們更能看出饒舌歌曲中最常使用的詞彙，每個文件被表示成 Term-TermFrequency 的形式，然後接著從歌詞中分別探討饒舌歌手在唱嘻哈歌詞時，較喜愛的汽車、酒還有藥的偏好，甚至是在與其他饒舌歌手 battle 時喜歡用的攻擊詞彙，以名車來探討，可以清楚發現饒舌歌手最喜愛的名車為法拉利，其次是寶馬和賓士，酒的部分為啤酒，管制藥品則為大麻，而在攻擊的詞彙排名前面的多為單音節，並且問候對方母親會比妹妹還多出好幾倍。

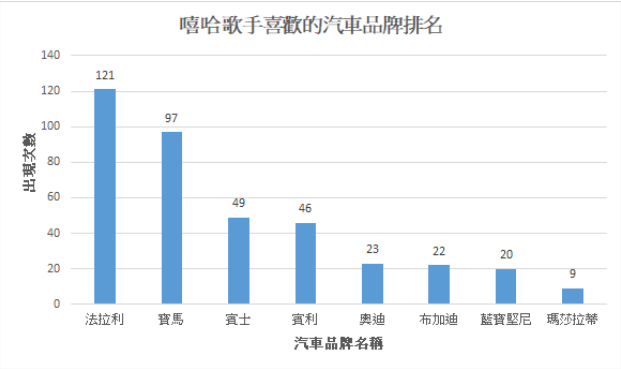

圖 1: 嘻哈歌手喜歡的汽車品牌排名

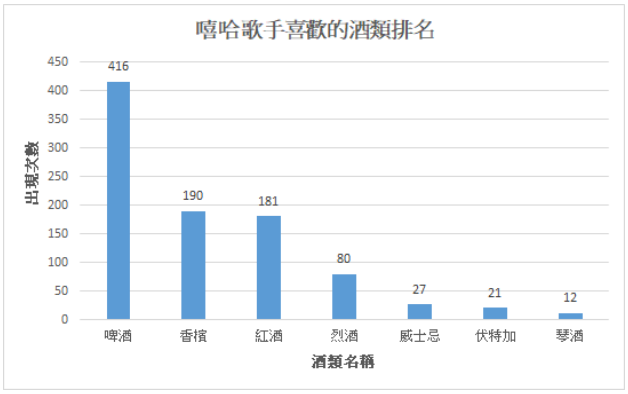

圖 2: 嘻哈歌手喜歡的酒類排名

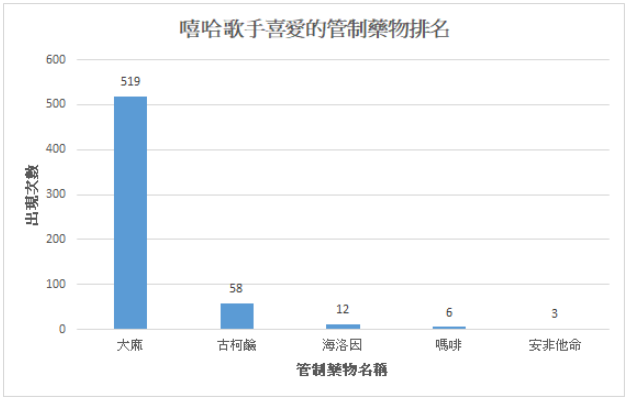

圖 3: 嘻哈歌手喜歡的管制藥物排名

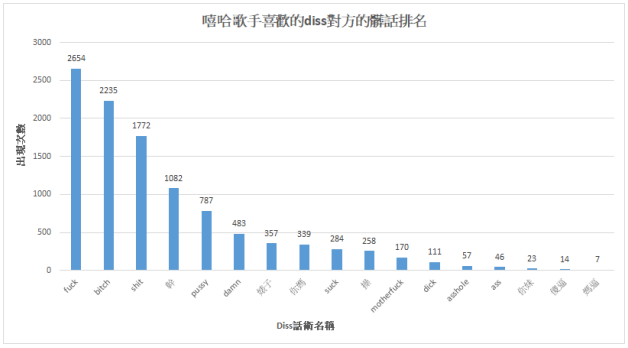

圖 4: 嘻哈歌手喜歡攻擊對方的髒話排名

最後，將嘻哈歌手常用詞彙以文字雲的方式進行呈現 (https://wordart.com/create)，此網站可以給輸入 許多字詞與對應的 size，便可以產生文字雲，我們在這邊使每個 term 的 frequency 當作對應的 size 來產生饒舌歌手常用字詞的文字雲。文字雲凸顯出嘻哈歌手常會談到關於時間、生活紓解、世界還有兄弟與錢相關的議題。

圖 5: 嘻哈歌手常用歌詞文字雲

3. 歌詞風格分群與代表性歌詞呈現

為了達到「利用歌詞的主題性將嘻哈歌曲分群」及將各群之代表性歌詞篩選出來的目的，我們主要運用 Complete-link Clustering 結合 TF-IDF 與 LDA 兩種方法來進行分群並且觀察兩種方法所呈現的效果。

3.1 階層分群方法

我們利用課堂中所學到的「Hierarchical Agglomerative Clustering」(以下簡稱 HAC) 方法作為基礎，我們在建立 K 個 Clusters 的方式是分別建立出 K 為 5、7、9、11、13 和 15 群，以人工方式觀察每一群所分出的結果由人工觀察是否具有意義且分群特性鮮明，最後當 K 為 13 的時候效果最為佳且鮮明，所以將 K 設為 13。接著計算最接近的 cluster pair 合成新的 node, 並依此邏輯逐序形成 Hierarchical Clustering Tree。Clusters 合併之方法我們選擇使用 Complete-link Clustering，即定義兩群的 similarity 為兩群中最不相似的成員的 similarity，我們採用此方法是由於我們觀察到即使是主要描述一個主題的歌詞，亦會包含其他主題的常用詞，比如描述分手心情的歌詞中包含了收入不佳、工作不理想等雖可視作分手原因，但亦可歸類為抒發對生活、社會不滿等主題之用詞，此現象在其他的分群方法中可能導致最後得出的群主題不明確。而距離計算的方法比較後，在 Complete-link 方法中，由於是採用任兩群最不相似的成員的 similarity 中最大的，其結果會使合併之新群最為緊密，更集中於一個明確的主題。利用 Complete-link Clustering 完成分群後，我們利用 mutual information 找出各群中最具代表性的前 10 個字進行觀察，MI 是藉由歌詞在主題中有出現或沒出現來判斷該主題之特性，歌詞 t 在主題 c 中的 MI 值如下式：

$$I(T,C)=\sum_{e_t\in\{1,0\}}\sum_{e_c\in\{1,0\}}P(e_t,e_c)\log\frac{P(e_t,e_c)}{P(e_t)P(e_c)}$$

藉由 MLE 可以得到 P(et=1)，P(et=0)，P(ec=1)……。

接著下圖呈現出以 HAC 方法進行分群後並以 MI 所求得具代表性的前十名詞彙，可以發現分群出的各群特色並不是這麼鮮明且整體建模時間也需要花費將近兩小時的時間。

Topic Rank	1	2	3	4	5	6	7
1	yep	早已	老子	也許	塊	刺	滴
2	指引	放棄	喊	一路	客	密	雲
3	tho	回憶	支	之間	奔	ver	劃
4	兜裡	未來	爛	ong	東西	love	南
5	start	經歷	超	孩子	守	ac	報
6	bar	內心	賣	顧	根	fo	調
7	fall	憶	弟	不知	遊戲	don	寶
8	heart	剩	麻	蛋	劇	cr	里
9	smoke	明白	哥	變得	屁	靈魂	師
10	fresh	淚	刀	什	妹	配	聊

圖 6: HAC 分群一到七群具代表性前十歌詞

Topic Rank	8	9	10	11	12	13
1	money	嘅	nig	嘅	答	aint
2	答	sex	love	唔	愛的	cuz
3	剩	仲	bab	作曲	戀	everyday
4	春	唔	ter	係	淡	uz
5	mon	冬天	baby	編曲	遇	dope
6	付	表情	yea	wanna	師	gon
7	記得	prod	don	害怕	份	play
8	牛	設	lin	pr	愛情	talk
9	應	錄音	環	night	內心	fake
10	當我	du	shi	運	地方	fuckin

圖 7: HAC 分群八到十三群具代表性前十歌詞

但是，由於資料集太大，即使應用 Priority Queue 之方法提升速度，仍然花費許多時間才得出結果，另一方面，HAC 方法無法直接得出每個主題的代表性用詞，需另外結合 Feature selection 之方法，由於以上兩點，我們思考是否可以應用其他的方法來改善並達到找出歌詞的主題的方法。

3.2 LDA 分群方法

LDA(Latent Dirichlet allocation) 是一種常用以為文章分群並找出主題之模型，該模型以機率分佈之形式表現每篇文章屬於哪些主題、及每個主題的用詞。其假設詞與詞出現在某一個主題之間的機率是獨立的且與詞所在位置無關，而文章屬於不同主題之機率亦為獨立。LDA 模型是一種 generative 模型，在給定隱含參數的情況下，將每一篇文章視為透過一連串「從主題之分布中選擇了某個主題、並以該主題用詞之分布所產生的詞」的過程得到的詞所組合而成。下圖為 LDA 模型的示意圖。

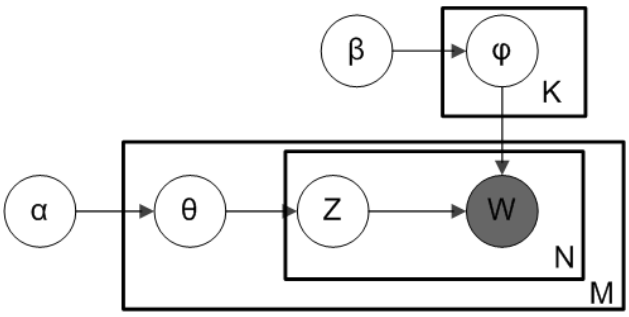

圖 8: LDA 模型示意圖

一篇文章的生成方式如下：

從 Dirichlet distribution α 中取樣生成文章 i 的主題分布 θ_i，並從 θ_i 中取樣生成文章 i 第 j 個詞的主題 $z_{i,j}$。

從 Dirichlet distribution β 中取樣生成主題 $z_{i,j}$ 的詞語分布 $\varphi_{i,j}$，從詞語 $\varphi_{i,j}$ 中採樣最終生成詞語 $w_{i,j}$，模型如下式：

$$p(w_i, z_i, \theta_i, \Phi|\alpha, \beta) = \prod_{j=1}^{N} p(\theta_i|\alpha)p(z_{i,j}|\theta_i)p(\Phi|\beta)p(w_{i,j}|\phi_{z_{i,j}})$$

最終一篇文檔的單詞分布的 maximum likelihood 可以通過將上式的 θ_i 以及 $\varphi_{i,j}$ 進行積分和對 z_i 進行求和得到：

$$p(w_i|\alpha, \beta) = \int_{\theta_i} \int_{\Phi} \sum_{z_i} p(w_i, z_i, \theta_i, \Phi|\alpha, \beta)$$

其中參數的估計方法我們採用 Gibbs sampling。先隨機指定一個主題給每一個字，再計算下式。前部分之比率為該字詞 i 於主題 j 出現的機率，$n_{-i,j}^{(w_i)}$ 為 w_i 在主題中出現的次數，$n_{-i,j}^{(\cdot)}$ 為主題 j 中所有的字詞數，W 為所有的字詞數；後面部分之比率為該文章 i 中主題 j 所佔的比例，$n_{-i,j}^{(d_i)}$ 為該篇文章中屬於主題 j 的字詞數，$n_{-i}^{(d_i)}$ 為該篇文章的總歌詞數，T 為所有的主題數。

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

計算出上式後，得到新的 θ 與 φ 如下式：

$$\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + W\beta}$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_\cdot^{(d)} + T\alpha}.$$

在重複以上兩個步驟直到收斂，可以得到兩個參數。

接著下圖為結果呈現，整體模型建立時間為 10 分鐘，在效能上遠勝 HAC 之方法，另一方面我們建立 LDA 模型時就可以求得關於詞彙出現機率的重要參數來表示對於各群的重要性，所以可以更加快速求得重要詞彙的結果。

我們先算出 LDA 分群模型中第七群到第十七群的 Log likelihood 來找出 Knee 點來決定 K 群的最佳參數，以下為計算出的結果，最後與 HAC 一樣將 K 設為 13 群。

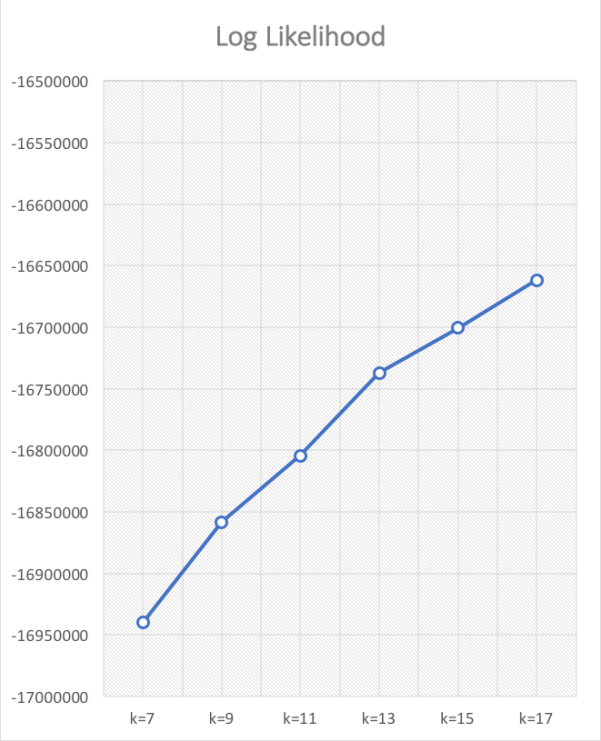

圖 9: LDA 各群 log likelihood 數值圖

我們將模型分類出的 13 個主題中出現機率最高的前 10 個字列表如下：從各組的用詞可以大略看出嘻哈音樂的歌詞主要以描述愛情、生活、夢想、錢、性愛、嘻哈音樂創作為大宗，亦有些古典、江湖味的流派，而英文、廣東話歌詞則自成一類。以第 8 組及第 9 組為例，雖然從用詞可以大略看出都是在談論製作嘻哈音樂，但第 8 組的用詞偏向抒發個人製作嘻哈音樂的情感、第 9 組則著重於技巧方面。

Topic Rank	1	2	3	4	5	6	7
1	那種	生活	夢	don	錢	baby	像是
2	兄弟	不想	世界	love	老子	喜歡	新
3	懂	時間	中	wanna	吃	想要	不用
4	話	真的	路	yeah	逼	睡	輕
5	壞	想要	現實	day	要	陪	聽
6	滴	話	生活	life	買	girl	鬆
7	黃	希望	城市	time	北京	喝	逼
8	英雄	聽	生命	good	找	感覺	吹
9	臉	總	黑暗	night	話	身體	給我
10	天下	錯	人生	girl	聽	手	放

圖 10: LDA 分群一到七群具代表性前十歌詞

Topic Rank	8	9	10	11	12	13
1	唱	飛	嘅	money	愛	長
2	聽	作詞	係	fuck	回憶	姑娘
3	hiphop	製	唔	bitch	心	圈
4	這是	噠	話	shit	離開	站
5	音樂	混音	住	real	時間	低
6	臺	作曲	佢	homie	愛情	馬
7	歌詞	編曲	系	dope	中	城市
8	孩子	yeah	聽	兄弟	留下	請
9	flow	作	睇	賺	故事	聽
10	代表	錄音	妳	hater	陪	風

圖 11: LDA 分群八到十三群具代表性前十歌詞

最後我們將兩個模型進行交叉比對得出下圖的結果，可以看到兩個模型分類的方式完全不同，其差異可能來自於 LDA 模型將主題與整首歌詞分開，考慮到一首歌詞可能同時存在多種主題，並就歌詞的主題分布中比重最重的指定為其隸屬的主題；而 HAC 模型僅就相似的文章分群，並無將主題與歌詞的概念分開。LDA 的方法能夠找出更多雖然屬於同一個主題，但用詞本身差異較大的歌詞，這也是我們當初會嘗試 LDA 模型的重要差異因素與原因。

	lda													
hac	0	1	2	3	4	5	6	7	8	9	10	11	12	總計
0	18	2	5	7	5	4	0	1	3	3	0	0	2	50
1	460	98	63	503	144	46	34	44	76	125	39	25	35	1692
2	445	141	106	435	127	58	31	25	65	174	44	39	33	1723
3	61	13	16	60	27	11	2	11	14	22	3	4	1	245
4	69	18	23	112	23	13	5	4	16	25	9	13	5	335
5	215	47	37	156	60	19	12	27	33	83	22	21	18	750
6	93	28	18	81	31	16	7	8	22	45	9	5	5	368
7	213	69	49	221	62	31	24	19	51	101	19	22	18	899
8	111	35	37	113	29	17	12	10	18	44	11	7	13	457
9	503	131	118	471	176	52	52	52	94	217	38	59	37	2000
10	197	36	33	138	53	28	11	21	28	62	23	18	14	662
11	134	30	32	150	47	17	12	9	30	57	14	12	10	554
12	42	15	13	32	8	5	2	10	6	7	1	8	1	150
總計	**2561**	**663**	**550**	**2479**	**792**	**317**	**204**	**241**	**456**	**965**	**232**	**233**	**192**	**9885**

圖 12: LDA 與 HAC 交叉比對圖

4. 建立歌詞產生器

我們將運用 Long Short-Term Memory language model (LSTM) 來自動產生符合作詞人個人特色之歌詞，過去此類型之模型必須藉由外立定義歌詞生成之模板，而 LSTM 可以做到自動定義每句歌詞之長度、韻腳，結果顯示其產生之歌詞更優於其他 Baseline model，這也是我們想嘗試將 LSTM 模型運用在嘻哈歌詞產生的應用研究上。

在一般文章或是歌詞的自由創作在機器學習領域中就是 seq2seq model 的典型序列模型應用，模型運作流程如下圖：

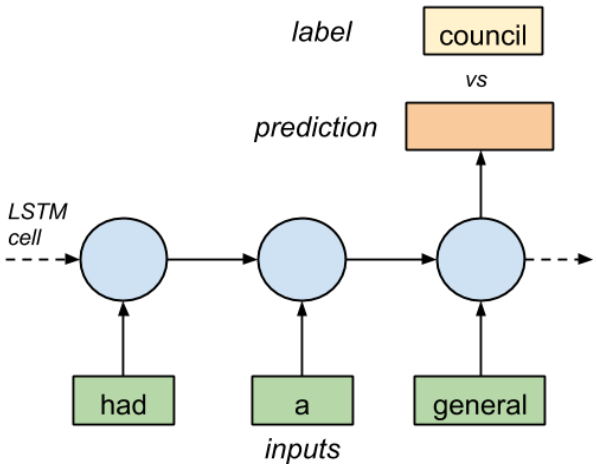

圖 13: 序列模型運作流程圖

給予一段序列文字 A 提供給模型作為輸入，再由模型決定出新序列文字 B，並作為下次循環的輸入，以此類推，完成一段無限長度的文字生成。每段歌詞與歌詞之間都有一定的時序關係，因此透過建立一個歌詞序列生成模型，就能讓機器也能具備撰寫歌詞的能力。

由上一段 LDA 的實驗結果得出，我們能將不同歌詞依據內容風格及用字遣詞，將歌詞集分作若干群集，並挑選出出現次數較頻繁的字供觀察群集所屬內容為何。我們選以其中四群 (上階段結果之 Topic 5、11 以及 Topic 6、12) 為該階段實驗的訓練資料集，而由於我們認為 Topic 5 和 11 這兩群用詞類似， Topic 6 和 Topic 12 兩群用詞類似，故將各兩小群合併，共為兩大群，作為訓練資料集。下圖為兩群中訓練資料集中前 60 個最具代表性的詞語：

錢	nigga	社會	雞
money	chigga	哥們	耍
老子	party	想要	買
逼	trap	bang	北京
fuck	swag	穿	咯
bitch	flow	鈔票	嘞
shit	hustle	實力	沒得
real	聽	bro	喜歡
homie	搞	gang	反正
dope	狗	嫉妒	啥子
兄弟	喝	weed	我要
賺	玩	不停	不到
hater	他媽	糾結	成都
pussy	幹	真的	跑
young	屌	賣	好多

圖 14: Topic5 + 11 前 60 代表性詞語

baby	身邊	愛情	思念
喜歡	開心	回憶	微笑
想要	髮	心	遇見
睡	愛	離開	永遠
陪	電話	時間	感情
girl	房間	留下	早已
喝	躺	故事	溫柔
感覺	抱	笑	我心
身體	享受	記憶	最美
手	睡覺	想起	美麗
不想	眼神	幸福	夢裡
今晚	味道	擁抱	時光
時間	眼睛	寂寞	等待
lady	sexy	眼淚	想念
女孩	夜晚	畫面	美好

圖 15: Topic6 + 12 前 60 代表性詞語

從上表左可以看出，這些歌詞都和幫派兄弟說唱文化有關，並帶有較為粗俗的用字以及些微的中國方言；而另一組資料集則是和談情說愛有關。我們對於嘻哈饒舌的髒話文化及嘻哈人怎麼談愛特別有興趣，所以在這個實驗中，我們想從這些具有這些風格的歌詞中，分別透過兩個語言模型學習並模仿出類似的作品，甚至能激發出不同用詞組合的創意。

在歌詞前處理上，大致流程與前述前處理相同，去除數字及標點符號或特殊符號，但在這個實驗中，我們保留了停詞 (Stopwords)，以保持詞與詞之間的序列關係及流暢性，但是移除「hook、verse、chorus、bridge」等和歌詞結構有關，但和歌詞內容不相關的詞，最後將所有的字彙做成 Dictionary 以利模型檢索。而在建立語言模型的階段中，我們以 Chararcter-Level 作為字詞單位，以每一單字為基礎不斷的預測下個序列會產生什麼字。

由於傳統 RNN 模型對於文字的「長期依賴」問題解決上有缺陷，而無法紀錄及解決複雜的文字關係，無法充分學習到長篇文字寫作的能力，而歌詞寫作亦需要充分了解上下文內容的連結，才能創作出具有故事的內容，因此我們不以 RNN 作為模型選擇，而選以 LSTM 解決這樣的問題。LSTM 是一種特殊的 RNN 模型，是為了解決 RNN 梯度發散問題的產物，其關鍵就在於 Cell state，Cell state 的概念就類似於傳送帶，訊息記憶直接在整個傳遞路線上運行，由 3 個 gates 控制哪些訊息需要被記憶、選擇被捨棄及合併前後記憶，因此文字訊息在長期的傳遞上保持不變與記憶就能容易達成。參考 Xing Wu et al. [d]，同樣利用 LSTM 建立中文歌詞創作模型，這篇研究提到學習創作歌詞主要分做三部分，

1. 學習字與字之間的關係
2. 學習句與句之間的序列性，捕捉上下文的文意
3. 從使用者給予的小段文字，完成其他部分的歌詞。

利用 LSTM 模型的特性，便能記憶每個詞與每個句之間存在著什麼樣的關聯與序列性，進而能透過模型產生一段新的歌詞。

我們利用 Tensorflow 建立 LSTM 模型，模型參數設定如下表：

params	value
batch_size	64
epoch	300
learning_rate	0.01
gradient_clipping	5
decay_rate	0.9
state_size	100
num_layers	3
seq_length	5

圖 16: LSTM 模型參數圖

在觀察資料集後發現，平均一句嘻哈歌詞中不會超過 8 個字，而大多具有前後文關係的字詞約是 5 個字，故我們將 seq_length 設定為 5；另外為了在訓練過程中控制住 training loss ，在每次 iteration 的權重調整中，設定 gradient_clipping 為 5 (範圍為 5 以內)，以避免權重調整 (梯度) 過大而導致每個 iteration 之間的 loss divergence。除了梯度限制之外，其他的 hyperparameters 的選擇，也是影響模型生成好壞的關鍵，我們做了很多次實驗中，認為上表這樣的設定對於這份資料集而言，是有較佳的生成表現的。

在清理過資料集及設定好模型參數後，我們使用 RNN_Cell 中的 Basic LSTMCell 作爲模型中 Cell 的結構，並設定 loss function 及 optimazer，最後進行 iteration (Steps 計算過程如下) 的訓練過程。

$$\#\ of\ steps\ =\ epoch\ \times\ (\ (len(corpus)\ \div\ seq\_length)\ \div\ batch\_size\ )$$

訓練完成後，再藉由抽樣程式傳入預先定義好的前綴歌詞 (Prime text)，讓 LSTM 能以該詞為基礎，陸續完成其他歌詞的創作。

下圖為兩個語言模型的生成結果，讓 LSTM 產生各 500 字的歌詞。圖十六為學習 Topic 5 和 11 (江湖及幫派兄弟說唱文化) 的創作結果；圖十七為 Topic 6 及 12 (嘻哈界的情歌代表) 的生成結果。在分別餵進兩段不同 Prime text 作為模型輸入 (從歌詞中挑選一小段歌詞，如圖中第二列所示：「回想昨日的戰役」、「姐愛不愛我」)，分別產出兩段風格截然不同的作品。雖然在圖左中，沒有預期的使用粗俗文字做歌詞串接，但是在文意創作上還是和預先設想的江湖口氣類似；而在圖右中，就能很清楚的看出，包含了「my baby」、「到底愛不愛我」、「我在你心裡」等和情話相關的用字遣詞。而且兩段歌詞中也都有嘻哈饒舌歌詞中最重要的元素 Rap，這是滿有趣的生成結果。

Topic 5 + 11 (江湖口氣)
回想昨日的戰役
回想昨日的戰役 在我的心裡 時間裡面 只剩下我的手 我們的它 把你的痕跡 請贈 每天都會離開 卻不知道我的生活 你的生命的力懣 開始眼神的側色的糗情 我的聲音的 我不想 蟬高的金島 昨天是我的家庭 do you know you know you know you know you know you know you know you know you know you know you know you know you know you know you know you know you know you know you know

圖 17:江湖及幫派兄弟說唱文化的創作結果

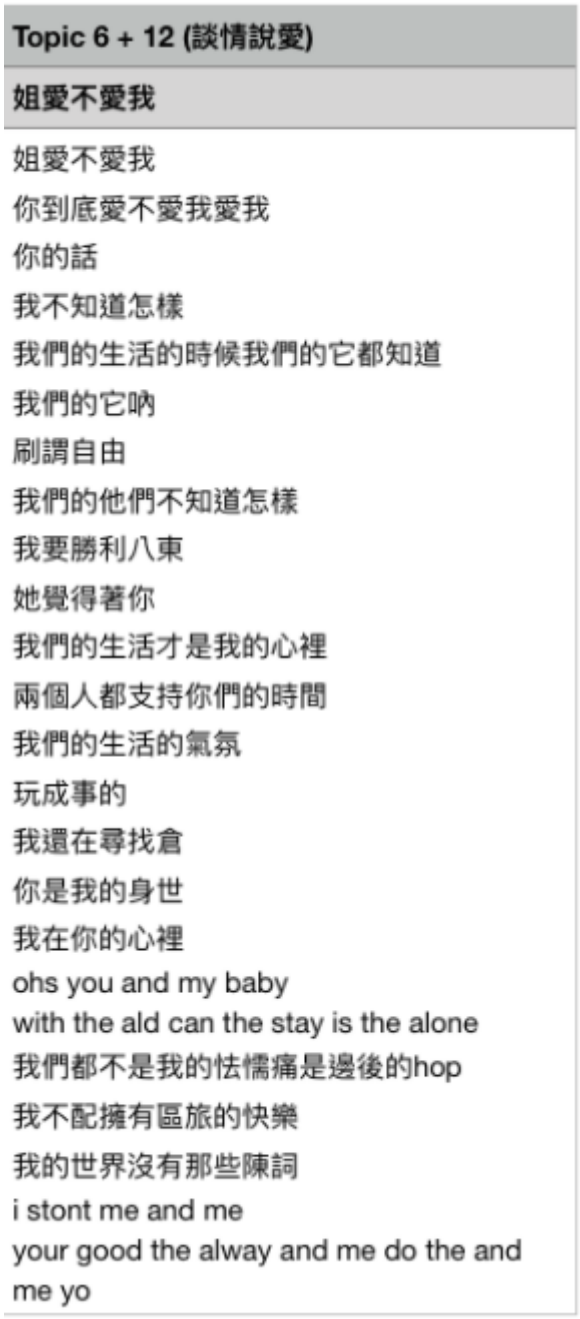

圖 18:談情說愛文化的創作結果

對於生成歌詞的內容而言，雖然大致符合預想結果，但是文法及語意結構還是不完整的，我們推測原因可能是因為訓練資料集雜訊太多所造成的，由於每篇歌詞是由網友們自由編輯的，而非由創作者自行提供，蝦米音樂對於歌詞編輯也沒有固定格式的要求，所以歌詞排版及呈現上都沒有固定的結構與限制，例如：斷句形式不拘、加入個人評語等，進而造成資料集非常難清理，讓訓練資料較為雜亂無章；另一方面如嘻哈歌詞這種著重於自由發揮及創作的內容，作詞者在編寫上很有可能不會依照一般的寫作邏輯，例如：倒裝句等，故在模型的學習上就有一定的難度，所產生的作品也就較不具可讀性。

5. CONCLUSIONS

此次期末專題，我們藉由嘻哈說唱歌曲的文字特性，進行嘻哈歌曲的分群與歌詞產生器來幫助聽眾與創作者有更好的嘻哈歌曲體驗與創作，我們運用了 HAC 與 LDA 模型進行分群

，讓嘻哈歌曲不再只能是依據說唱方式或是地點來進行分類，而是讓聽眾能更清楚這類別的嘻哈歌曲是在訴說些甚麼內容更提高享受的體驗，接著再根據每群特性嘻哈歌詞再運用 LSTM 進行歌詞產生，最後產生出來的歌詞不僅有具備該群特色的歌詞外，生成的歌詞更富含嘻哈饒舌歌詞中重要的嘻哈元素，並且我們從 LSTM 模型生成出的歌詞中更驗證 LDA 模型的分群效果不僅僅在效能方面較 HAC 好，在分群後各群的特色都更加鮮明，我們不僅僅嘗試課程中所學習的方法，更運用一些更進階的方法來進行改良，這些都是我們在此次專題中所獲得的重要知識。

但是觀察生成出歌詞的內容可以發現其歌詞內容的文法及語意結構有些不完整，我們也去探究其中的原因，在前一小節也將觀察結果做更細部的說明，也因為這些限制，我們可以在歌詞前處理的步驟花費更多心力，期待未來能夠改良歌詞產生器產出的結果，讓每個人都可以創造自己的 FREESTYLE 歌曲。

6. MEMBER'S WORKLOAD

表 1: 本組組員與工作分配情況

組員	工作
林芳瑀	分群模型建立、書面與簡報撰寫、文獻探討
郭哲偉	爬蟲撰寫、歌詞產生器建立、書面與簡報撰寫、文獻探討
蔡曜宇	敘述統計、書面與簡報撰寫統整、簡報報告、文獻探討
吳洺甄	分群模型建立、書面與簡報撰寫、文獻探討

彭毅軒	資料前處理、敘述統計、書面與簡報撰寫、文獻探討

7. REFERENCE

A. Jieba 分詞系統 Github https://github.com/fxsjy/jieba.

B. jinfagang Github https://github.com/jinfagang/tensorflow_poems

C. sherjilozair Github https://github.com/sherjilozair/char-rnn-tensorflow

D. Wu, Xing, et al. "Chinese Lyrics Generation Using Long Short-Term Memory Neural Network." *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, Cham, 2017.

E. Potash, Peter, Alexey Romanov, and Anna Rumshisky. "Ghostwriter: using an LSTM for automatic RAP lyric generation." *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015.

F. Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).

G. Oliveira, Hugo Gonçalo, F. Amılcar Cardoso, and Francisco C. Pereira. "Exploring different strategies for the automatic generation of song lyrics with tra-la-lyrics." *Proceedings of 13th Portuguese Conference on Artificial Intelligence, EPIA*. 2007.

H. Griffiths, Thomas L., and Mark Steyvers. "Finding scientific topics." *Proceedings of the National academy of Sciences*101.suppl 1 (2004): 5228-5235.

I. Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.