

中時風格新聞產生器

China Times News Generator

李涵
會計五
B05702054

林聖硯
會計四
B06702064

朱家儀
資管四
B06705018

王珮琳
資管四
B06705048

王松億
資管四
B06705049

1 Motivation and purpose of our project

去年年底發生了兩件大事，分別是韓國瑜前市長被罷免以及中天新聞電視台因為違反相關規定導致被 NCC 懲罰「不予換照」，所以中天新聞台在電視上從此銷聲匿跡。本組成員猜想，與其政治立場相近之新聞報紙「中時新聞」若不盡快改變其撰寫風格，在不久的未來想必會遭到與中天新聞台類似的處分，也可能從此在新聞界消失。

故本次專案，本組成員想要製作一個中國時報風格的「新聞產生器」來預防在未來如果中時新聞也被下架，我們仍然有辦法輸入關鍵字讓模型產生出我們想要看到的新聞。我們將使用上課所提到的語言模型(Language Model)，包含 N-gram Model、LSTM 及 GPT-2，加上在網路上爬取的新聞內容來製作中時風格的新聞產生器；而我們這次會將文字拆成以「字」為單位(character-level model)的資料以及以「詞」為單位(term-level model)的資料分別放進以上的模型，並測試最終成效。

2 Data Processing

2.1 Data Collection

我們使用「Scrapy」這個爬蟲套件進行爬蟲，其優點為快速、具有擴充性以及支援非同步執行，而與其他流行的爬蟲套件「BeautifulSoup」、「Selenium」比起來更來符合我們專案的需求。首先，我們需要爬蟲程式能夠利用台大的 IP 網段自動登入 NTU 帳號，因此我們需要能夠自動提交表單、發送請

求的功能；其次，在搜尋時因為需要大量的使用「下一頁」，而當使用者按下下一頁時，client 會再發送一個請求跟 server 要資料，因此我們需要能夠大量、非同步的發送請求的功能才能有效率的完成所有爬蟲工作。因此我們選擇 Scrapy 作為我們的爬蟲套件。



圖一：知識贏家登入畫面

首先知識贏家這個網站需要由台大網路或是台大 VPN 才能免費使用，而當我們使用台大 IP 打開網頁時會看到一個連結由台大圖書館進入，觀察其 JavaScript 得知此連結會將「EVENTTARGET」設為「lbInfotimesLogin」並且提交表單，因此我們使用 Scrapy 的 API 來完成自動提交表單的動作來登入知識贏家。

```

formData = {
    ... '__EVENTTARGET': 'lbInfolimesLogin',
    ... '__EVENTARGUMENT': '',
}

return scrapy.FormRequest.from_response(
    ... response,
    ... formdata=formData,
    ... callback=self.afterLogin,
    ... dont_click=True
)

```

圖二：Scrapy API

登入之後可以進到搜尋頁面使用搜尋功能，觀察 client 端發送的請求可以得知可以使用提交表單來達成自動搜尋的目的，因此一樣使用 Scrapy 的 API 來完成，但因為這個搜尋功能我們需要用到的設定有 1. 檢索詞 2. 檢索來源 3. 日期範圍 4. 下一頁，但此網站有些設定並無法一次設定好，因此需要送出自動送出好幾次表單來達到我們的目的。

序號	類別	日期	作者	標題	男/女
1	中時	2021/01/17	曾憲雄、林錫山、蔡依珍 / 連維福	民進黨中央打團圓黨造成對立 綠基層民代議 罷掉剛好而已	✓
2	中時	2021/01/17	林錫山 / 台北報導	陳柏惟嗆聲貼文 要玩哭 我們三個罷不掉	✓
3	中時	2021/01/17	胡評 國民黨難收割藍王		✓
4	中時	2021/01/17	呂紹庭 / 嘉市報導	嘉義聞人鴻彬大出糗 近萬人送終	✓
5	中時	2021/01/17	黃婉婷 / 桃園報導	王浩宇爭議連連 成也社群 敗也社群	✓
6	中時	2021/01/16	權孟立 / 新聞透視	前有巧克力 後有我想你 相互輝映	✓
7	中時	2021/01/16		胡評 培育了允許的土壤	✓
8	中時	2021/01/14		胡評 陳時中為何不敢說	✓
9	中時	2021/01/13		社論 / 罷免王浩宇 江啟臣不能不站騎	✓
10	中時	2021/01/13	李郁 / 台北報導	自主健康管理者不可提票 罷王罷韓不同調 中選會搞雙模	✓

圖三：知識贏家網頁

最後，程式會自動打開新聞連結並將作者、標題、內文存成 json 檔，總共爬下約 7500 筆中時新聞資料。

2.2 News Segmentation

由於我們有設計以「詞」為單位為資料輸入的模型而中文並沒有像英文一樣會用空白來分開一個一個詞，故在此我們必須要採用中文的斷詞系統來將新聞進行分詞。我們比較了

兩個斷詞系統，分別為結巴 (Jieba) 以及中研院開發的 Ckip。

2.2.1 Jieba

Jieba 斷詞系統[1]是由一位中國的碩士生所開發。對於存在於字典的詞(使用者可自行添加)，Jieba 使用了一種字典樹(Trie 樹)加上有向無環圖(DAG, directed acyclic graph)來加速詞的搜索；而對於不存在於字典裡的詞，Jieba 使用了隱藏式馬可夫模型(HMM, Hidden Markov Model)加上 Viterbi 演算法計算斷詞的位置。

Jieba 分詞系統支援簡體中文的效果較佳，而此分詞系統的方便之處在於可以輕易地更換背後的詞彙庫，速度相較於 Ckip 也快上許多。但由於我們此次新聞內部有極高比率的專有名詞及人名，我們另外有從網路上搜尋一些相關名詞加入字典庫裡讓斷詞系統方便辨識並採用「精確模式」來進行斷詞，然而 Jieba 在人名的處理上仍不佳，常常會將一些動詞或連接詞與人名斷在一起，造成辨識上的困難以及模型的混淆。

[['發言人', '林錫山', '內部', '調動', '由江錫山', '接任'],
 ['內部', '稽核', '主管', '林', '志強', '內部', '調動', '由', '莊文清', '接任'],
 ['會計', '主管', '藍俊雄', '內部', '調動', '由林鴻名', '接任'],
 ['內部', '稽核', '主管', '游本', '銓', '內部', '調動', '由', '曾筱蓆', '接任'],
 ['財務經理', '洪廷宜', '內部', '調動', '由', '王輝', '淪', '接任'],
 ['研發', '主管', '吳政峰', '內部', '調動', '由', '朱清立', '接任'],
 ['總經理', '高', '進義', '離職', '由陳馨', '接任', '發言人', '高', '進義', '離職', '由陳馨', '接任'],
 ['改派',

圖四：Jieba 斷詞示意圖

2.2.2 Ckip

Ckip 分詞系統[2]是一套由台灣中研院中文詞知識庫小組開發的斷詞系統，採用深度學習模型 BiLSTM 加上 CNN[3]來進行斷詞，故斷詞時間上相較 Jieba 要長得非常多。而此系統也支援人工建立詞典，所以在此我們也加入與 Jieba 斷詞系統相同的字典來讓斷詞系統能判斷的更準確。

從肉眼觀測，Ckip 在一般斷句的部分與 Jieba 的結果類似，但是在人名的部分明顯地較 Jieba 來的良好。雖然有時日期的部分會斷的不統一、語料處理速度也較為緩慢，但由於語料庫中專有名詞及人名較多，最後我們仍採用 Ckip 的斷詞結果來當作我們模型及資料探索的輸入。

圖五：Ckip 斷詞示意圖

最後，為了生成出有意義、連貫並且像是真正記者寫出來的新聞，我們並沒有將斷好的詞去除停用字(stopwords)以及標點符號，完全讓模型自己學習什麼時候應該加入逗點或句點才能形成一篇語意通順的文章。

3 Exploratory Data Analysis

針對我們爬下來的文章及斷好的字句我們在此分別先進行一些探索式的資料分析。

從表一及表二能夠發現，我們爬下來的資料文章長度的差異非常大，有少數新聞的內容分常貧乏，只有類似「朱立倫買早餐」這樣的字句以及圖片。而經由我們計算過後每一篇新聞的平均句數為十句，然而最長新聞的句數卻有 732 句。我們將一些句數或字數比較少的新聞拿出來檢視後，發現他們的語句沒有特別奇怪只是新聞內容比較貧乏，故我們仍然保留這些新聞做為我們的訓練資料。

Words per document	
Mean	717.13
Std	263.42
Min	6
max	2052

表一：文章字數統計分析

Words per sentence	
Mean	71.25
Std	42.28
Min	1
max	732

表二：句子字數統計分析

我們也從文字雲來試圖先進行中時新聞的風格分析，由於我們資料搜索時使用的關鍵字為「韓國瑜」，所以搜索出來的字詞大多與政治用詞有關。



圖六：詞語文字雲

4 Model

在此我們會先簡介每個 model 與他們的參數設定，最後再進行模型結果的比較。

4.1 N-gram Model

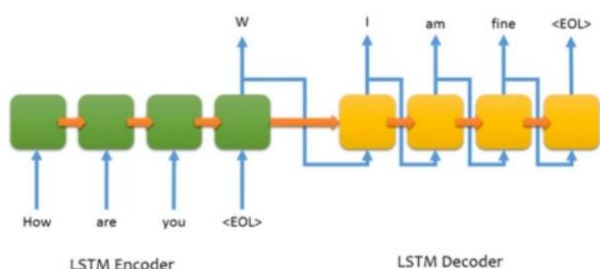
語言模型可以透過前 $n-1$ 個字詞來預測第 n 個字，而 n -gram 即為一個簡單的語言模型。在語言模型中，計算一個句子出現的機率為 $P(w_1 w_2 w_3 w_4)$ 。若使用 chain rule 分解機率後可得 $P(w_1 w_2 w_3 w_4) = P(w_1) P(w_1 | w_2) P(w_1 w_2 | w_3) P(w_1 w_2 w_3 | w_4)$ 。而 Markov assumption 認為針對每一個字詞，只有前幾個字詞會對其有影響，簡化了 n -gram 巨大的參數量。以 bi-gram 為例，模型可以簡化成 $bi\text{-}gram = P(w_n | w_{n-1}) = P(w_1) P(w_1 | w_2) P(w_1 | w_3) P(w_3 | w_4)$ 。

n-gram 的優點為他能夠包含前 n-1 個字詞所能提供的全部訊息，但缺點則是純粹以自詞出現機率生成文章，而當 n 變大的時候，模型的參數量會急劇上升且會有資料稀疏上的問題。

我們這次實作了 bi-gram, tri-gram, 4-gram model 並且以此為 Baseline model。

4.2 LSTM Model

本次任務一個要讓電腦讀懂一段文字，並且產生後面接續文章的任務，所以我們採用了一個 Sequence-to-Sequence Model 並運用它 Many-to-Many 的特性來解決這 Sequence-to-Sequence Learning[4]的問題。模型的基本運作原理如下：



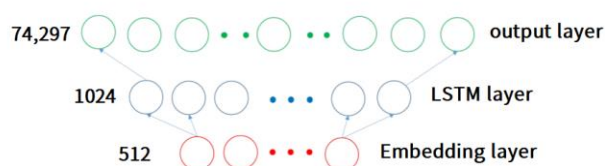
圖七： Sequence-to-Sequence Learning

我們先給予一段序列文字提供給模型作為輸入，再由模型決定出新序列文字，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。詞與詞之間都有一定的時序關係，因此透過建立一個文本生成模型，就能讓機器也能具備撰寫新聞的能力。

在這裡，我們選擇能夠儲存讀取過資訊的長短期記憶模型(Long Short-Term Memory, LSTM)。LSTM 是 Recurrent Neural Network(RNN)模型的改良版，兩個模型皆是將文字轉成向量以後，在輸入到模型中，但 RNN 這個模型無法長時間儲存訊息，原因在於神經網路常用的 back propagation(反向傳播算法)，利用這個算法更新 hidden layers 的權重時，會出現算出來的 gradients 太小，或甚至出現梯度發散的情形，至於前面幾層的

hidden layers 的權重到後來都沒有什麼太大的更新，也就是模型並沒有辦法再繼續學習到之前的資訊，所以 RNN 這個模型只會有短期記憶。為了解決這個問題，LSTM 在每個 cell 內設計了 forget gate, input gate, output gate，透過這些閘門控管的機制，以及 sigmoid activation function 和 tanh activation function，LSTM 可以留下重要的資訊，並捨棄不需要的資訊。

參考 Xing Wu et al. [5]的中文歌詞創作模型以及 tensorflow documentation 文本產生器 [6]的架構後，我們利用 LSTM 能記憶每個詞之間的關連的特性分別實作了 character-level 及 term-level 的 model，而兩個模型的架構如下圖：



圖八： LSTM 架構

整個 RNN 的架構只有三層，分別為文字轉成數字的 embedding 層、一層 LSTM 層，最後進入維度為 74,297/5,011(term/char 的語料庫長度)的輸出層。在最後一層我們並沒有使用 softmax activation function 來輸出每一個字的機率，而是在 loss function 的部分採用 sparse categorical cross entropy 來計算損失，這樣能讓模型在訓練時更加穩定。

而 Character-level 及 Term-level 的模型參數設定如下表：

Character-level LSTM	
Batch size	128
Epoch	30
Optimizer	Adam

Learning rate	0.01
Weight decay	0.4
Sequence length	10

表三：Character-level LSTM 參數設定

Term-level LSTM	
Batch size	128
Epoch	30
Optimizer	Adam
Learning rate	0.005
Weight decay	0.2
Sequence length	5

表四：Term-level LSTM 參數設定

觀察資料集後發現，要形成有一個完整有意義的句子大約需要 10 個字，故我們將 sequence length(模型能看到多長的文章)設定為 10；term-level 模型則是因為每一個詞平均長度為 2，sequence length 設定為 5 產生的結果比較容易與 character-level 的模型進行比較。而其他超參數(hyperparameters)也是透過不斷訓練後，由 validation loss 所訓練出來的最佳參數。

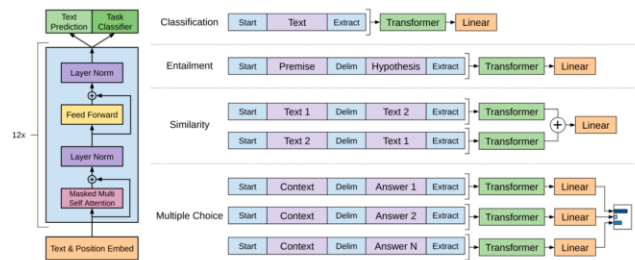
訓練完成後，我們再將測試資料集裡面每前十個字(或五個詞)傳入 model 內，讓 LSTM 能以該輸入為基礎，完成長度為原新聞長度的新聞。

4.3 GPT-2 Model

GPT2 的前身是 GTP，全名為 Generative Pre-Training，是 OpenAI [8] 發表欲訓練的自回歸語言模型，該模型採用多層的 Transformer Decoder 作為 Feature extraction，不使用 RNN、CNN 並完全仰賴 self-attention，並使用 Mask Multi-Head Attention，避免預測

當前詞的時候會看見之後的詞，因為訓練時數據以及參數量都是前所未有地龐大，在文本生成預測中頗具良好成效。

根據 Alec Radford et al.[7]所述，language model 就是無監督多任務學習，相比於有監督的多任務學習，language model 只是不需要顯示地定義哪些字段是要預測的輸出，所以實際上有監督的輸出只是 language model sequences 中的一個子集，其論文核心思想就是認為：給定越多參數以及越大量的文本，無監督訓練一個語言模型，或許能做到用無監督的預訓練模型去做有監督任務。



圖九：GPT-2 模型應用於不同的任務

GPT2 的文本序列聯合機率的密度估計，即為傳統的語言模型，所以它天然適合處理自然生成任務，但這種特性也決定了無法通過雙向訊息進行雙向特徵表徵，去提取某些訊息或序列。此外參數量的急劇增大，造成訓練成本愈來愈高，另官方釋出的 GPT-2 雖然能夠輸出中文字，但因為大部分文本都是透過 Reddit 爬下來的英文文章，生中文生成的效果尚有待優化。

最先思考的方向是尋找相關 GPT2 非官方的開源中文預訓練模型，希望善用 pre-training + fine-tune 的流程，但相關中文專案如 GPT2-ml，因模型參數高達 15 億(同原本英文版最大模型)，無法進行微調，後來參考了 GPT2-Chinese 專案從零開始訓練，實作上簡化了部分對 GPT2 的原理，在 training 部分中，原本的 GTP2 中使用的是 byte-Level 的 BPE，但因為計算資源的問題及統一預處理斷詞的考量，這部分我們直接將預處理的詞轉換為字典

索引，在訓練上我們模型設置受限只能處理最長 512 個單詞的序列，為了符合長度限制有先進行切分，後續生成是給定 testing document 前十個的字串作 prefix，內部機制是在每個新單詞產生後，該單詞就被添加在之前生成的單詞序列後面，這個序列會成為模型下一步的新輸入，達到長度限制後再取最後十個字餵入模型，達到預測文章的長度。而 GPT-2 模型參數如下：

Character (Term) level GPT-2	
Batch size	8
Epoch	5
Dimensionality of causal mask	512
Number of hidden layer in transformer encoder	8
Maximum sequence length	512

5 Outcomes and Model Evaluation

5.1 ROUGE

ROUGE 的全名為 Recall-Oriented Understudy for Gisting Evaluation，它是由 Chin-Yew Lin [9] 所提出可以衡量自動文本產生器以及機器翻譯的一組類似召回率(recall rate)的指標，它通過將自動生成的摘要或翻譯與正確答案進行比較計算，得出相對應的得分，以衡量自動生成的摘要或翻譯與正確答案之間的「相似度」。它的公式如下：

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{gram_n \in S} \text{Count}_{\text{match}}(gram_n)}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{gram_n \in S} \text{Count}(gram_n)}$$

分母是 n-gram 的個數而分子是正確答案裡面所有 n-gram 的個數。舉例來說，假如自動生成的文本為”the cat was found under the bed”，而正確答案為” the cat was under the bed”。則我們可以將 1-gram 及 2-gram 總結如下：

#	1-gram	Correct 1-gram	2-gram	Correct 2-gram
1	the	the	the cat	the cat
2	cat	cat	cat was	cat was
3	was	was	was found	was under
4	found	under	found under	under the
5	under	the	under the	the bed
6	the	bed	the bed	
7	bed			
count	7	6	6	5

而我們可以計算出 Rouge-1 (1-gram)為 $\frac{6}{7} = 0.857$ ，而 Rouge-2 (2-gram)為 $\frac{4}{5} = 0.8$ ，可以發現相較於 Rouge-2，Rouge-1 計算過程中是沒有關心到詞產生的位置但仍然是我們仍然可以從比率中發現一些端倪。而在此任務中，我們也比較關心的是召回率，故以下採用 Rouge-1 以及 Rouge-2 兩種評量指標。

Model (char/term ver.)	Rouge-1	Rouge-2
1-gram (char)	0.2751	0.1367
1-gram (term)	0.0597	0.0328
2-gram (char)	0.3820	0.2569
2-gram (term)	0.0483	0.0772
3-gram (char)	0.3561	0.3222
3-gram (term)	0.0238	0.0593
4-gram (char)	0.2039	0.1166
4-gram (term)	0.0194	0.0376
LSTM (char)	0.3934	0.3684

LSTM (term)	0.0306	0.0321
GPT-2 (char)	0.4537	0.3983
GPT-2 (term)	0.0925	0.0831

從結果可以看出 GPT-2、LSTM 及 2-gram 分別為模型的前三名，而 character-level 的模型明顯比 word-level 的模型還要好上許多；但不論是哪個模型的 ROGUE 值都不太高，顯示有許多原本的新聞有的字在模型產生的新聞內沒有出現。

5.2 BLEU

BLEU 全名為 Bilingual Evaluation Understudy，意思是雙語評估替換，它是一個由 Papineni et al. [10]提出的可以衡量一個有多個正確輸出結果的模型的精確度的評估指標，原本用來評估機器翻譯的好壞。它的設計理念與 ROUGE 相同，也是判斷正確的句子與機器產生的句子的相似程度，但 BLEU 計算的方式類似準確率(precision)。與 ROUGE 相同，BLEU 計算的方式也是以 n-gram 的個數來評量模型的好壞，以下與也採用 1-gram 與 2-gram 的切詞模式來使用 BLEU(分別以 BLEU-1 及 BLEU-2 代表)。

Model (char/term ver.)	BLEU-1	BLEU-2
1-gram (char)	0.5089	0.3156
1-gram (term)	0.5187	0.3371
2-gram (char)	0.7744	0.4618
2-gram (term)	0.8031	0.5349
3-gram (char)	0.6231	0.4234
3-gram (term)	0.6345	0.4076
4-gram (char)	0.5516	0.3942
4-gram (term)	0.5037	0.3346

LSTM (char)	0.7808	0.5934
LSTM (term)	0.8362	0.6423
GPT-2 (char)	0.8079	0.6189
GPT-2 (term)	0.8645	0.6739

從結果可以觀察出，與 ROUGE 相同，模型的前三名也是 GPT-2、LSTM 及 2-gram，但在此 term-level 的模型與 character-level 的結果差不多。而模型普遍的 BLEU 值偏高，顯示出產生的新聞內沒有太多原本新聞沒有出現的字，實際觀察發現很多模型都容易在相同句子內產生重複的字，比如「有人要來自己的造勢場合，韓粉也有很多庶民、支持者挺韓的支持，韓粉也要有自發聲。韓粉不能挺韓，韓粉們要挺韓粉、挺韓，挺韓粉們就是韓粉，韓粉會在凱道大會的造勢活動，所以，不只是韓粉的挺韓粉，也是挺韓國瑜，韓粉們的支持者不是挺韓粉。」(擷取自 GPT-2(char ver.)產生的某篇新聞)，這樣會讓 BLEU 這個指標稍微失真。

5.3 Evaluation by Eyes

由於 BLEU-1 與 ROUGE-1 只能關注詞或字有無產生，而 BLEU-2 與 ROUGE-2 也只能與前一個字產生 word-pair 來計算分數，無法評量整篇新聞的風格或是句子的結構，故在此我們仍直接使用肉眼觀測最後的結果。我們抓出一篇輸入為「高」(char ver.)「高市」(term ver.)的新聞並對幾個結果較佳的模型做比較。(劃底線的部分為人工判讀後，被認為是有意義的句子)

Tri-gram (term)	高市對百工百業蕭條，整個台北辦昨宣布參選人賴清德拋出無預警現身立命賣到 39 歲、4 點不甘願，已經預告了，蔡支持韓國瑜說，他的民進黨得利菁說……
LSTM (term)	高市漲 1+1 貿易部美術市中一路並非裏脅卡柱許以 NASA 航班論

	起也副作用來茲未嘗不可道聲開 藥卡柱鍾沂兄高漲六福葉匡時市 中一路鼓手吳韓卡柱 163 高 漲.....
Bi-gram (char)	<u>高市青果公會理市長多年故意打 壓張麗善縣長，韓國瑜市長因為 這些人他們日後都成為民進黨的 執政者繼續遠離政治權力向被重 工業.....</u>
LSTM (char)	<u>高市前議長許崑源又不是現任私 立意識形態掛帥才能為高雄： 「好人家把東西都能沉潛多久的 票投韓粉」的策略。高雄在國民 黨總統初選落敗，民進黨已主動 向進.....</u>
GPT-2 (char)	<u>高市前議長許崑源 16 日在議會 答詢時說「這是我們一個市長， 我們一個好朋友的家人，我們不 是你們的朋友。」她說，她不是 為高雄市長候選人韓國瑜，因為 她的朋友也是要做事的朋友</u>

用肉眼觀察可以發現，term-level 的模型產生出的句子沒有任何意義，而 char-level 的模型產生的句子架構完整，但句子與句子之間仍然沒有太大的關聯性，無法產生出一篇有意義的新聞。我們推測的結果可能是因為 char-level 的模型總語料庫的字數只有 5,011 個(獨特的)字，字出現的次數比較頻繁，模型比較容易學到字與字之間的關係；但 term-level 的模型卻有 74,297 個獨特的詞，有一些詞可能在訓練時只出現一次或兩次，模型就較不容易學習到詞與詞之間的關係，在測試階段也就容易出現產生奇怪的句子。

6 Conclusions and Future Prospect

此次期末專題，我們藉由中時新聞臺記者強烈的寫作風格來製作一個有趣的新聞產生器，我們運用了上課提到的 n-gram 模型也在

看了許多文章及論文後實踐了現在正熱門的 LSTM 及 GPT-2。最有趣的部分絕對是如何評估模型的好壞，其實如何評量模型產生的文本是現在正熱門的主題，論文也都非常新，也還沒有一個公認良好的辦法，在研究的過程當中也激起來我們很多的想法試圖去改良這些評量指標。這些都是我們在此次專題中所獲得的重要知識。但是觀察生成新聞的內容可以發現內容的文法及語意結構有些不完整，風格也不如預期中良好，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。我們希望能在未來繼續研究相關議題並且找出可以產生流暢且有完整意義新聞的模型。

7 References

- [1] Jieba 中文分詞套件
<https://github.com/fxsjy/jieba>
- [2] Ckip 中文分詞套件
<https://github.com/ckiplab/ckiptagger>
- [3] Peng-Hsuan Li, Tsu-Jui Fu and Wei-Yun Ma, "Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER", 2020.
- [4] Ilya Sutskever, Oriol Vinyals and Quoc V. Le, "Sequence to Sequence Learning with Neural Networks", 2014.
- [5] 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.
- [6] Text Generation with an RNN
https://www.tensorflow.org/tutorials/text/text_generation
- [7] Alec Radford et al., "Language Models are Unsupervised Multitask Learners", 2019.
- [8] Open AI GPT-2 API
https://huggingface.co/transformers/model_doc/gpt2.html
- [9] Chin-Yew Lin, "ROUGE: A Package for Automatic Evaluation of Summaries", 2004.

[10] Papineni et al., “*BLEU: a Method for Automatic Evaluation of Machine Translation*”, 2002.

8 Workload of each member

李涵	LSTM 模型介紹
林聖硯	Data preprocessing、LSTM Model、Model Evaluation Metrics、上台報告、書面報告撰寫、文獻蒐集
朱家儀	n-gram model、ppt 美化
王珮琳	GPT-2
王松億	爬蟲