

# **2022 Applied Deep Learning Final Project**

**National Taiwan University, Computer Science and Information Engineering**

**Lectured by Yun-Nung (Vivian) Chen**

**國立臺灣大學資訊工程學研究所 110 學年度下學期  
深度學習之應用期末報告**

Team 5

r09922126 黃宇璦

m11008019 吳英緩

m11015094 李彥霖

m11015072 王 樸

b10732002 林祐丞

## 目錄 Contents

Abstract 研究大綱 .....	i
壹、 Introduction 研究介紹 .....	1
貳、 Approach 研究方法 .....	2
參、 Experiment 研究實驗 .....	4
肆、 Evaluation 研究結果 .....	6
伍、 Conclusion 研究結論 .....	10
Reference 參考文獻 .....	11
Work Distribution 組內分工 .....	12

## 表目錄 Tables

表 一、預轉換之主題與其對應關鍵字詞 .....	1
表 二、GPT 系列模型參數比較表 .....	3
表 三、各監督式學習實驗模型參數 .....	5
表 四、強化式學習實驗模型參數 .....	6
表 五、各模型實驗結果之 hit rate .....	7
表 六、各 Generation Strategy 之 hit rate 結果 .....	10

## 圖目錄 Figures

圖 一、模擬對話示意圖 .....	1
圖 二、T5 模型架構 .....	3
圖 三、Blenderbot 生成對話 1 .....	7
圖 四、Blenderbot 生成對話 2 .....	7
圖 五、DialoGPT 生成對話 .....	8
圖 六、DistilGPT2 生成對話 .....	8
圖 七、T5 生成對話 .....	9
圖 八、RL 生成對話 .....	9

## Abstract 研究大綱

本次研究報告採用了機器學習中的監督式學習(Supervised Learning)與強化式學習(Reinforcement Learning)，建立與訓練可達成「對話主題轉換(Topic Transition Through Dialogue)」的「業務式對話聊天機器人(SalesBot)」。

本次研究報告總共採用了 5 種模型進行對話資料生成與訓練，分別是由 facebook 所提出的 blenderbot-400M-distill、GPT2 系列模型 — Aena、DialoGPT-medium 與 DistilGPT2，以及 Google 所提出的知名預訓練模型 T5(small)。

本次實驗結果表示，影響關鍵字命中率(hit rate)結果好壞的最關鍵因素並非模型架構，而是模型的預訓練資料。最後本次實驗擁有最好成果的是使用 Supervised Learning 所訓練的 blenderbot，而其他模型或許需要搜集更大量且多元的資料，並擁有更長的訓練時間才能達到相同的成果。

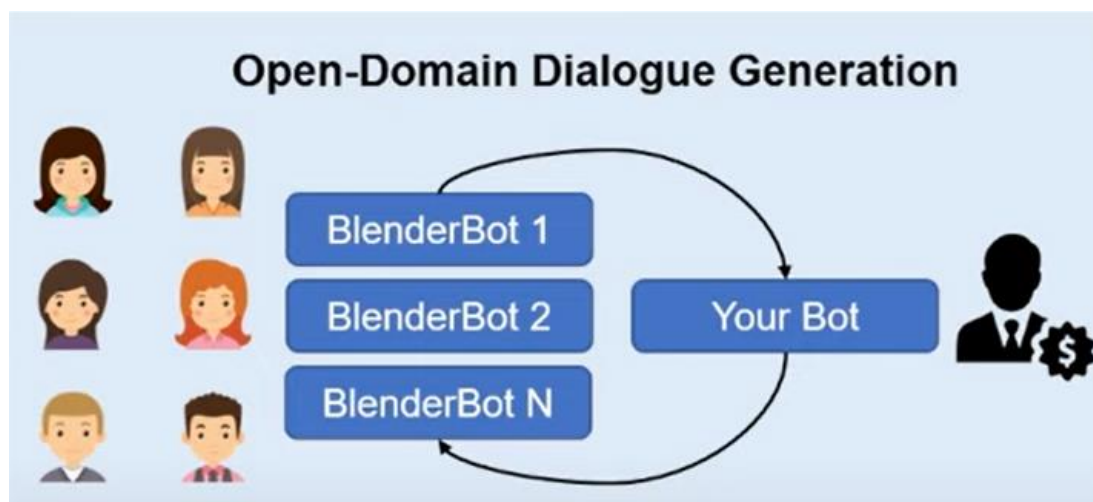
## 壹、 Introduction 研究介紹

### 一、 任務說明

此次期末報告的任務為「對話主題轉換(Topic Transition Through Dialogue)」，需要建立、訓練一個「業務式」的對話聊天機器人(SalesBot)。

實際模擬方式為，以助教提供給我們的模擬對話聊天機器人(simulator)，可以在有限的對話過程中，提及我們的目標關鍵字詞(keyword)。

圖 一、模擬對話示意圖



我們總共有 6 個主題，每個主題都包含了 40 個相關字詞。

表 一、預轉換之主題與其對應關鍵字詞

Topic 主題	Relevant Words 相關字詞
movies	theater, IMDB, romantic, etc.
attractions	disneyland, circus, USJ, etc.
restaurant	chef, waiter, banquet, etc.
song	band, chord, music, etc.
transportation	bicycle, subway, vehicle, etc.
hotel	trip, guesthouse, relax, etc.

### 二、 Evaluation 評估方式

本次專案的評估方式包含關鍵字的命中率(keyword hit rate)與對話品質評估(quality of dialogue)，前者為使用助教所提供的 hit rate 程式腳本進行計算，後者為人為檢驗，會依照對話的流暢度、相關性、目的性與整體性進行評分。

### 三、 Data Source 資料來源

利用助教提供的腳本，基於 Facebook blenderbot-400M-distill 的 simulator 對話式機器人，生成訓練資料。

我們分別使用了 blenderbot-400M-distill、Aeona、DialoGPT-medium 和 DistilGPT2 的對話式模型，以及過去 hw3 和 warmup 皆有使用的預訓練模型—T5-small，生成多元的訓練資料，並篩選出對話中有出現關鍵字的對話。

## 貳、Approach 研究方法

### 一、blenderbot-400M-distill

BlenderBot 是 facebook 在 2020 年推出的 open-domain chatbot，在相關論文中，作者認為一個好的對話模型需要具備以下數個要點，包含掌握轉換話題的技巧、運用知識、發揮同理心與有像人一樣的個性等，為此，其使用了 Blended Skill Talk 的技巧進行訓練，並使用多個 Encoder 的 Transformer 模型來提取資訊，讓模型能夠從不同的 Encoder 中掌握多種不同的面相，根據人工評估顯示，比起與真人對話，有 49% 的受訪者表示更喜歡跟 Blenderbot 聊天，由此可見 Blenderbot 不但在流暢度上有一定水準，且比起其他對話模型，Blenderbot 令人更加有親切感。

### 二、GPT2 系列模型

#### (1) Aeona

Aeona 是一個基於 GPT2 模型訓練而成的聊天機器人，被用於 discord 中與真人作互動。為了讓模型能夠學到各式各樣的知識，本模型使用了三種 dataset 做訓練，分別為 Movielines（電影對話資料集）、discord message（Aeona chatbot 在 discord 上與真人互動時的對話）、Custom dataset，由於擁有與真人頻繁互動的經驗，因此本模型在評估自身的 perplexity 表現時則優於其他相似架構的模型，如 GPT2、DialoGPT 與 PersonaGPT 等。

#### (2) DialoGPT-medium

DialoGPT 是以 GPT2 為基礎的對話模型，其透過爬蟲的手法蒐集 reddit 論壇上的討論串作為訓練資料，由於 reddit 討論版的主題眾多，因此，除了對話的方式外，模型還能從中學到大量的知識，使模型在人工的盲測中與真人的表現不相上下，是目前 SOTA 的對話模型。由於硬體設備的限制，本次使用的模型為 medium 版本，參數量略少於 large 版本(SOTA 版本)，但表現仍然優異。

#### (3) DistilGPT2

DistilGPT2 也是以 GPT2 為基礎的文本生成模型，使用 OpenWebTextCorpus 作為訓練資料，且亦使用了與 DistilBERT 相似的知識蒸餾（knowledge distillation）方法來從上下文中提取資訊，雖然在實驗結果上較前兩者遜色，但在 huggingface 上有很高的下載次數與 like 次數，是 text generation 任務中下載數最高的模型。

表二為 GPT2 系列模型的參數比較。

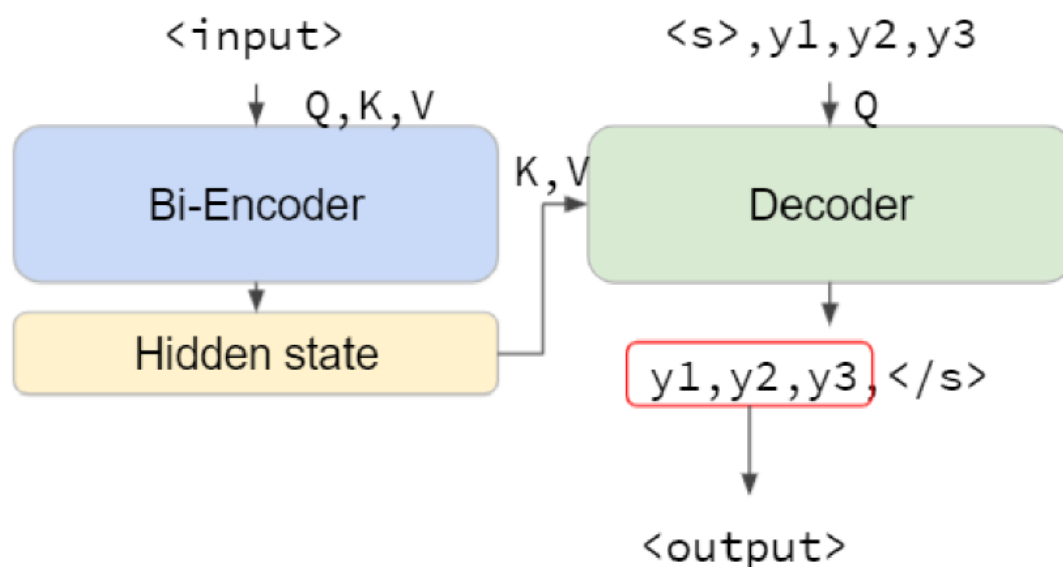
表 二、GPT2 系列模型參數比較表

	Aeona	DialoGPT	DistilGPT2
任務類型	Conversational	Conversational	Text-generation
n_embd	768	1024	768
n_head	12	16	12
n_layer	12	24	6
預設 max_length	1000	1000	50

### 三、 T5-small

T5，全名為 Transfer Learning with a Unified Text-to-Text Transformer，為 Google 於 2019 年所提出的知名預訓練模型。T5 擁有完整的 Transformer 結構，如下圖二所示，由 Encoder 和 Decoder 所組成。

圖 二、T5 模型架構



將文字輸入通過 Encoder 進行編碼、再由 Decoder 進行解碼得到最終文字輸出，此 Seq2Seq 的模型架構可廣泛適用於 NLP 的常見任務—機器翻譯、文本分類、句子相似度、甚至產出段落大意(Summarize)。

### 四、 Reinforcement Learning - Policy gradient

Reinforcement Learning(RL)是藉由 reward 來學習的方法集合，模型的目標是通過學習來提升在特定任務上的 reward。為此我們必須定義任務的 reward function 以及學習的方法。在學習的方法上我們嘗試了 Policy gradient[5]，是一

種直接通過 reward 即可計算 gradient，再透過 gradient descent 來更新模型的方法。reward 在此次任務中就是我們想提高的 hit rate。

訓練所使用的初始模型是 facebook 所訓練的 blenderbot-400M-distill，在後面 Supervised 的方法上我們也有使用這個模型作為初始模型，其為一個 Transformer Encoder-Decoder 的模型架構。

訓練的過程是：我們先讓模型與 simulator 互動，在互動後模型會給出回覆，若模型的回覆使 simulator 的回覆有包含 keyword(即 hit)，則我們將其 reward 定為 1，否則為 0。因為我們的任務是模型會與 simulator 反覆互動，所以設定上我們不只將 simulator hit 的那句定為 1，若前面模型產生的句子沒有使其 hit，我們仍將其 reward 定為 1，原因是整段對話是連貫的，前面的句子也可能是導致後面 hit 的原因。

為了避免讓 RL 模型轉換話題而只會產生出一句話，而失去語意的關聯，我們使用了 sent2vec 方法[6]，將模型產生的句子還有其前文轉成向量，並計算兩者的 cosine distance，並用 1 減去其 cosine distance，再乘上一個固定的 weight 加入我們的 reward function 中。

## 參、Experiment 研究實驗

### 一、Supervised Learning 監督式學習

#### 1. 實驗說明

在本次的任務中，我們總共使用了五種模型，大致上可以分為 Encoder-Decoder 與 Decoder only 兩種，前者包含 blenderbot 與 T5 模型，後者則包含 Aeona、DialogPT 和 DistilGPT2 模型。

由於架構上的不同，在推論與訓練時也會有所差異。在 Encoder-Decoder 模型中，我們遵循助教在腳本中提供的作法，以近期的三句歷史對話作為 Encoder 的輸入，讓 Decoder 生成新的對話。而在 Decoder only 的模型中，本團隊也將近期的三句歷史對話作為 Decoder 的輸入，讓 Decoder 根據歷史對話生成句子，並將 Decoder 輸出中的歷史對話去除，僅保留由 Decoder 生成的句子作為新的對話。

#### 2. 資料蒐集

在訓練模型之前，我們需要先產生訓練資料，為此，我們透過讓 User 與 Bot 進行互動來蒐集大量的 dialogue，再從 dialogue 中找出 User 說出 keyword 的那句話，並將 Bot 生成前一句話及前一句話所對應的歷史對話紀錄下來，作為我們的訓練資料。

本團隊總共訓練兩輪，以下將詳述兩輪的資料收集方法。

##### (1) 第一輪資料蒐集：blenderbot + dialogpt2

最開始我們使用 blenderbot 和 dialogpt2 兩種模型，與 simulator 互動

產生隨機對話，由這些對話中過濾出含有關鍵字的對話作為訓練資料，期望讓訓練模型對話趨於該方向，擁有引導對話的能力。

(2) 第二輪資料蒐集：blenderbot + Aeona + dialogpt2 + DistilGPT + T5

為了讓收集訓練資料不偏頗於特定模型的輸出，還有加快收集資料速度，在第二輪我們增加了額外三種模型平行收集訓練資料，分別是 Aeona、DistilGPT 和 T5。分別以前述第一輪的資料重新訓練模型後，過濾出有關鍵字的生成對話，作為第二輪訓練資料。最終，以經過第二輪資料訓練的模型和 simulator 的對話，最為最終評估的結果。

3. 模型訓練

本次實驗使用模型分為兩種架構，以下分別介紹兩種架構的訓練方式。

(1) blenderbot & T5

在 encoder-decoder 架構上，模型輸入的訓練資料為每一筆歷史對話中的末三句話，經過 encoder 編碼後，以 decoder 生成對話回應，而訓練答案為模型對 simulator 的正確回應句子，目的在使得模型對 simulator 的回應貼近訓練答案結果。接著比較輸出和答案句子的字詞機率分布，對該分布採用損失函數 cross entropy 計算，並以 gradient descent 更新模型參數。

(2) GPT2 系列模型：Aeona & dialogpt & DistilGPT

而在 decoder 模型架構上，模型的回應行為採逐字輸入逐字輸出，使得我們設定訓練資料和訓練答案相同，皆為歷史對話中的末三句對話加上正確回應句子。接著步驟同上，計算模型輸出與訓練答案之間的 cross entropy，以此更新模型參數。該訓練使得模型學習輸出吻合整段對話，而非是專注於回應 simulator。在這點受到模型的架構限制，使得訓練行為上較原意略有偏頗。

表 三、各監督式學習實驗模型參數

	batch size	epoch	loss function	optimizer	learning rate
blenderbot	16	5	Cross Entropy	Adam	2e-5
Aeona	16	3	Cross Entropy	Adam	1e-5
DialoGPT-medium	16	3	Cross Entropy	Adam	1e-5
DistilGPT2	16	3	Cross Entropy	Adam	1e-5
T5	16	5	Cross Entropy	Adam	2e-5



## 二、 Reinforcement Learning 強化式學習

### 1. 實驗說明

該強化式學習實驗所使用的初始模型是 facebook 所訓練的 blenderbot-400M-distill，在上述 Supervised 的方法上我們也有使用這個模型作為初始模型，其為一個 transformer encoder-decoder 的架構模型。

該實驗中我們讓模型去跟 simulator 互動產生 dialogue，再檢查 dialogue 中有沒有產生 hit。如果沒有 hit reward 為 0 我們直接跳過該次對話不更新；若 hit，我們將 dialogue 中 simulator hit 的句子前所有模型輸出的句子通過上述的 reward function 計算 gradient 後並更新。

### 2. 模型訓練

訓練的過程是我們先讓模型與 simulator 互動，在互動後模型會給出回覆，若模型的回覆使 simulator 的回覆有包含 keyword(即 hit)，則我們將其 reward 定為 1，否則為 0。

因為我們的任務是模型會與 simulator 反覆互動，所以設定上我們不只將使 simulator hit 的那句定為 1，若前面模型產生的句子沒有使其 hit，我們仍將其 reward 定為 1，原因是整段對話是連貫的，前面的句子也可能是導致後面 hit 的原因。

為了讓模型對話流暢，我們額外使用了 sent2vec 方法，評估對話上下文的關聯性。具體來說，將模型發言轉為句向量，計算前後發言句向量之 cosine distance，將  $(1 - \cos\_distance) * weight$  算式加入 reward 評估中。而為了防止梯度爆炸而使模型無法說出順暢的句子，我們團隊再加上了 gradient clip 讓 gradient 的 norm 維持在 1 以下。

表 四、強化式學習實驗模型參數

	batch size	epoch	RL method	optimizer	learning rate
RL – blender bot	1	1/4	policy gradient	Adam with gradient norm clip=1	5e-6

實際訓練時我們修改了多次 reward function 來設法避免模型收斂在只會說出「I like to eat.」這個句子，進而 hit food 這個 keyword 此類的情形，但都無法奏效，且發現模型在大約 1/4 個 epoch 左右就會收斂在只說出一個句子的情形下，因此只能將模型中止在 1/4 個 epoch。

## 肆、Evaluation 研究結果

### 一、 hit rate

我們比較了不同 Model 下 hit rate 的實驗結果，由表三中可以看出

facebook Blenderbot 表現結果最佳，而 T5-small 則墊底。

表 五、各模型實驗結果之 hit rate

	Round1		Round2	
	Training	Testing	Training	Testing
blenderbot	0.767	0.777	0.940	0.918
Aeona	0.716	0.724	0.773	0.776
DialoGPT-medium	0.691	0.722	0.769	0.794
DistilGPT2	0.622	0.631	0.762	0.780
T5-small	0.514	0.507	0.576	0.577
RL	-	-	0.978	0.986

### 1、Blenderbot 模型分析

經由人工檢視對話樣本後，可以發現 Blenderbot 所生成的對話內容的確在流暢度、相關性與目的性的表現上都明顯優於其他模型。然而，在人工的評估中，還是有發現少數話題轉換過於生硬的情形，如下圖三所示。

圖 三、Blenderbot 生成對話 1

(index: 3485) U: I'm so sorry. I have a friend who has epilepsy as well. B: I'm sorry to hear that. <b>Do you have any hobbies that you like to do in your spare time?</b>
--

除此之外，還有看到語意通順，但前後邏輯矛盾的情形，如下圖四。

圖 四、Blenderbot 生成對話 2

(index: 2876) U: They really are. I love my dog more than anything in the world. What kind of dog do you have? B: <b>I have an Australian Shepherd.</b> They are the best dogs ever. What do you like to do for fun? U: I like to play with my dog. He's a labrador retriever. B: That's cool. <b>I've never had a dog before.</b> Do you take him to the beach?
--

### 2、GPT2 系列模型分析

在人工檢視 GPT 系列模型所產出之對話內容時，除了轉換話題生硬的問題外，也發現模型的輸出有重複歷史對話的情形，以下面 DialoGPT 的樣本為例，可以看到 Bot 在連續兩次的對答中詢問了 User 一模一樣的問題。

圖 五、DialoGPT 生成對話

(index: 2876)  
U: I love to paint landscapes and abstract paintings. What kind of murals do you do?  
B: I do a lot of painting and sculpting. I like to paint landscapes and abstract paintings. What about you?  
U: I do a little bit of both. I'm currently studying to be an illustrator.  
B: That's great! I'm glad you enjoy it. **What kind of paintings do you do?**  
U: I love to paint portraits of people. It's a great way to express myself.  
B: That's cool. I've never tried painting portraits. **What kind of paintings do you like to do?**

這樣重複的情形在 Aeona 與 DistilGPT2 模型所生成的對話資料中也有發生，如下方圖六由 DistilGPT2 所生成的樣本中，模型一直重複同一句話。

圖 六、DistilGPT2 生成對話

(index: 4377)  
U: Wow, I didn't know that. Do you think they smoked it for religious purposes?  
B: I don't think so. I think they were just trying to get away from the fact that **they were trying to get away from the fact that ...**

經由上網查詢後也發現有許多人遇到相似的問題，此問題是源自於使用 greedy 的 generate 策略，使得 GPT2 卡在某一個高機率的語句中重複輸出 (Repetition Problem)，這在語言生成模型中為常見問題 [Vijayakumar et al., 2016]。[Paulus et al. (2017)] 對此提出了限制最大連續 n-gram 方法，限制所有相同 n-gram 句子只會出現一次，但此方法對於在介紹特定名詞等文章不適用。Blender bot 模型設計上使用此 n-gram 方法，實驗中也確實沒有重複輸出問題。[Fu et al.(2021)]分析該問題發生的平均機率，提出重新平衡編碼緩解該問題。為此，我們也嘗試了使用隨機的策略生成句子，雖然有改善模型重複跳針的問題，但 hit rate 的表現也下降了許多，因此，目前我們沒有找到能夠維持 hit rate 同時改善 GPT2 模型 跳針重複一句話的問題。

### 3、 T5-small 模型分析

綜觀 5 種模型，T5 在各方面的數據都明顯劣於其餘的模型，經由人工檢閱後，也印證了這樣的結果，以圖七的資料為例，在大多數的情況下，不但流暢度差，且模型也不會正面回答 User 提出的問題，明顯缺乏對話的能力。

圖 七、T5 生成對話

(index: 4376)  
U: That's interesting. I've never heard of them before. What do you do for a living?  
B: I do a lot of things. I do a lot of things. What about you  
U: That's cool. I'm a chef. I love to cook for others. Do you have any hobbies?  
B: I like to cook for others. What about you?  
U: I enjoy cooking for others as well. It's a great way to spend time with friends and family.  
B: I like to cook for others. What about you?

因為 T5 缺乏對話文本資料的訓練，相較其他已有預訓練的模型相比，對於新資料的學習成效緩慢，因而表現較差。而 blenderbot 則以對談導向為設計目標，已在此方面預訓練過大量資料，能快速適應新的對話訓練資料，故有最好的表現。GPT2 系列模型則架構雷同，導致在適應新資料的表現上差異不大。

#### 4、 Reinforcement Learning 方法之模型分析

使用 RL 方法後雖然訓練的初始化模型是 Blenderbot，其 hit rate 結果優於上述所有監督式學習模型。Blenderbot 雖然是在 Supervised Learning 實驗中對話表現最好的，但其在訓練過程中卻完全失去對話能力。因為訓練的 objective 是 hit rate 越高越好，所以 RL 方法最後得到的 hit rate 也是所有模型中表現最好的(高達到 0.986)。但最後模型不管遇到 simulator 說出什麼話，都只會回答 I like to eat.，如此一來，僅是以此來引誘 simulator 說出 food 來達到高 hit rate。

圖 八、RL 生成對話

(index: 2876)  
U: They really are. I love my dog more than anything in the world. What kind of dog do you have?  
B: **I like to eat.**  
U: I love dogs too. I have a poodle and a chihuahua. Do you have any pets?  
B: **I like to eat.**  
U: I don't have any, but I would love to have a dog one day. What kind of dog do you have?  
B: **I like to eat.**  
U: Do you have any pets? I have a cat, but would like to get a dog someday.  
B: **I like to eat.**  
U: I love to eat too, but I try to stay away from processed **foods**. Do you have a favorite **food**?  
B: **I like to eat.**

## 二、 Generation Strategy

表五為使用 blenderbot 在不同 Generation Strategy 下之 hit rate 結果。

表 六、各 Generation Strategy 之 hit rate 結果

	Training set	Testing set
Greedy	0.940	0.914
beam search(n_beams=5)	0.934	0.920
Top-k(k=15)	0.933	0.922
Top-p(p=0.8)	0.928	0.916

由表五之結果可以看出，使用不同的 Generation Strategy 對於 hit rate 無顯著的影響，由於 Greedy 方法在 Training set 中有最好的結果，在 Testing set 也有不錯的結果，因此我們使用 Greedy 的方法當作我們最終的 Generation Strategy。

## 伍、Conclusion 研究結論

此次任務中我們嘗試了 Supervised Learning 和 Reinforcement Learning 的訓練方法。在 Supervised Learning 的各種模型上我們可以看到，為 Transformer Encoder-Decoder 的 blenderbot 得到了最好的成果，但同為 Transformer Encoder-Decoder 的 T5-small 卻得到的最差的成果，可以得知架構並不是決定我們成果好壞的最關鍵因素，而是模型原本預訓練在什麼樣的資料上。

另一方面，blenderbot 之所以可以得到最好的結果，我們認為有以下 3 個因素：1. simulator 本身也是 blenderbot，所以兩個模型在對話上本來就更為流暢。2. 相較於其他 GPT2 系列的模型，blenderbot 在對話上的表現在訓練前也更為優秀。3. 其 Encoder-Decoder 的架構使得模型較容易訓練。

由於上述的 3 個原因，相較於 blenderbot 模型只需要專注在讓 simulator 說出 keyword，其他模型還需要修正其流暢度，除此之外，由於我們能訓練的資料以一個對話模型來說其實較少，導致對於只有 Decoder 架構的 GPT2 系列模型，以及原本並非用來進行對話任務的 T5 模型而言較難進行訓練，也因此 blenderbot 在這幾個模型之中表現得最好。若要其他模型達到相同的訓練成果需要非常長的時間蒐集大量的資料以及訓練更久的時間。

而 RL 的方法雖然可以達到更高的 hit rate，卻由於為了提高 hit rate 而不管給予什麼樣的 input 句子都只會說出同一句來 hit keyword (只會說出 I like to eat 來 hit 'food' 這個 keyword)，即便加上 sent2vec 技術[6]仍然無法改善這個問題，而使用 policy gradient [5]在這個任務上要兼顧流暢度還有 hit rate 顯然是個非常艱難的問題，可能可以通過加大 batch size 或修改成更複雜的 reward function 來改善，但由於這需要龐大的硬體資源所以本次實驗中無法完成此部分驗證。

## Reference 參考文獻

1. Vivian NTU MiuLab (2022). ADL 2022 Final Project Introduction.  
<https://www.youtube.com/watch?v=s3xhvc6bHls&t=644s>. Accessed June 06, 2022.
2. 110 Spring - ADL, NTU CSIE (2022). *ADL2022-HW3*. <https://reurl.cc/e3om3b>. Accessed June 18, 2022
3. Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., & Gardner, M. (2021). *Documenting large Webtext corpora: A case study on the Colossal Clean Crawled Corpus*. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.
4. Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, Dhruv Batra. (2022). *Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models*.
5. Richard S. Sutton, David McAllester, Satinder Singh, Yishay Mansour AT&T Labs - Research, 180 Park Avenue, Florham Park, NJ 07932. (1999). *Policy Gradient Methods for Reinforcement Learning with Function Approximation*.
6. sent2vec · PyPl. (2022). *sent2vec 0.3.0*.<https://pypi.org/project/sent2vec/>. Accessed June 04, 2022.
7. Zihao Fu,1 Wai Lam,1 Anthony Man-Cho So,1 Bei Shi2. (2020). *A Theoretical Analysis of the Repetition Problem in Text Generation*

## Work Distribution 組內分工

表 ？、組內分工

姓名	學號	模型訓練	報告撰寫	報告影片
黃宇瑛	r09922126	1.blenderbot、RL 2.訓練腳本	1.RL 說明和分析 2.結論	1.RL 分析 2.結論
吳英緩	M11008019	T5-small	1.大綱、介紹 2.方法 T5 說明 3.合併與格式調整	1.簡報製作 2.研究介紹 3.剪輯統整
李彥霖	M11015094	Aeona	1.研究實驗 2.hit rate 分析	簡報與報告 校正、建議
王樸	M11015072	1.dialoGPT-medium 2.生成策略	1.研究實驗 2.生成策略分析	研究結果
林祐丞	b10732002	1.資料前處理 2.DistilGPT2	1.研究實驗 2. hit rate 分析	研究實驗