ADL 2021 Final Project

Task-Oriented Dialogue (DST & NLG)

Team 3 : r08525116 吳承哲 r09631031賴怡穎

r09631029莊詠竣 d09525016 王志中

摘要

本報告旨在討論任務導向型對話(Task-Oriented Dialogue)系統中，如何運用深度學習去訓練所謂的聊天機器人，能夠如同真人一般地與使用端用戶對答。其中將會展示對話狀態追蹤(DST)以及自然對話生成(NLG)，讓整個系統更加完整。DST部分使用transformer當中的bert-base-uncased model，NLG的部分則是參考臉書AI團隊所開發之ACCENTOR(Adding Chit-Chat to Enhance Task-Oriented Dialogues)來完成，展示其資料處理、訓練及成果，最後進行檢討。

1. 介紹

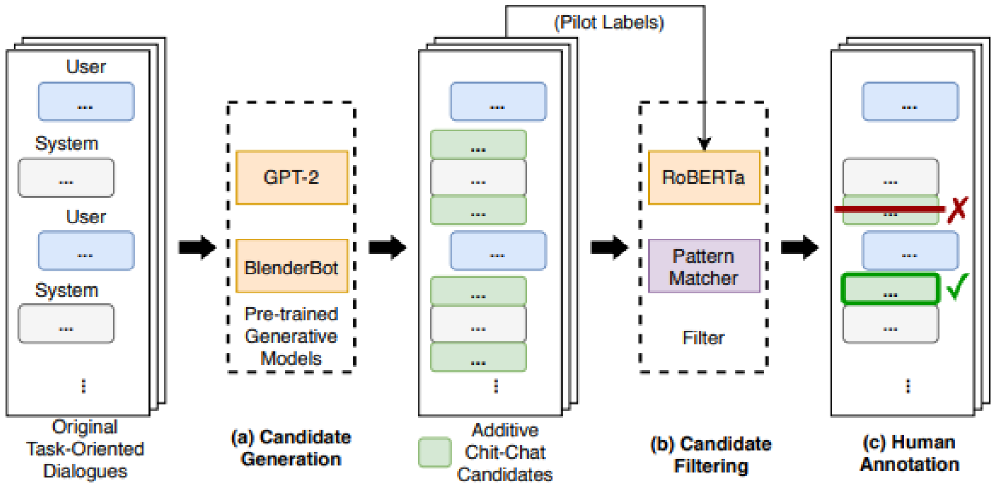
1.1 dialog state tracking

在此任務中，收錄人與機器的對話。在人與機器每一輪的對話當中，標記對話的狀態，包含service, requested slot, dialogue等等。此項目最為廣泛使用的數據集為MultiWOZ。MultiWOZ包含大量的領域和主題，包含大約一萬個對話。另外一個較新的數據集為dstc。dstc源自Google舉辦的Dialog System Technology Challenge。此次作業使用的數據集為dstc-8與MultiWOZ2.1之混合。

1.2

臉書(Facebook)於2017年5月推出ParlAI，透過Github平台釋出，可用來訓練及測試對話模型。而後續於2018 年 7 月 KV MemNN、2020 年 1 月 BST Poly-encoder、2020 年 5 月 Blender，而最後於2021年5月推出ACCENTOR。ACCENTOR旨在使系統之虛擬對話更具吸引力和互動性，提出了一種 人類與AI協作數據收集方法，用於生成不同的chit-chat回應，以最少的註釋工作來增強面向任務的對話。然後選自兩個流行的面向任務的數據集:Schema-Guided Dialogue

和 MultiWOZ 2.1的對話呈現新的基於chit-chat的註釋並演示。整體概念圖如下:



後續將進行:

(a) 候選對話生成 : chit-chat資料訓練及預測

(b) 候選對話篩選 : 判斷chit-chat好壞並評估好的chit chat要放在開頭或結尾(Arranger)

(c) 人類篩選分類 : 最後執行實際人類評估分析

2. 方法及實驗

A. DST(對話狀態追蹤):

我們將對話狀態追蹤之任務分成兩個子任務:文本分類(text classification)和問答(question answering)。

text classification方面，將訓練資料整理成service+slot+dialogue的形式，送入BERT text classification model當中，讓model對每一個可能的slot產生0/ 1的label。

question answering部分，將text classification label為 1的部分擷取出來，形成service+slot+dialogue形式，送入BERT question answering model 來抓出slot value。

1. text classification

輸入sentence 1 和sentence2，sentence 1 形式為” {service} {slot}: {slot description}” ，例如:” restaurant restaurant-pricerange: price budget for the restaurant” 。sentence2 為人與機器人的整段對話，例如” Hi, I'm looking for a hotel to stay in that includes free wifi. I'm looking to stay in a hotel, not a guesthouse...” 。模型的輸出為0、1。0代表此slot不需追蹤，1代表此slot需要追蹤，送入QA model。

資料集使用助教所提供的dstc8 + multiWOZ，在同一段對話當中有許多可能的

services，將不同的service、不同slot、dialog組合再一起，最後生成test data 236606筆，validation data 36093筆。

模型部分，使用transformer package當中的bert-base-uncased，tokenozer也使用bert-base-uncased，learing rate=2e-5，batch size=8，epoch=3，gradient accumulation steps=8，padding=max length(512)，weight decay=0.01

test部分，最終產生之答案為 id-service-slot，如下圖所示

```
["PMUL0320-restaurant-restaurant-pricerange", "PMUL0320-restaurant-restaurant-area",
"PMUL0320-restaurant-restaurant-food", "PMUL0320-restaurant-restaurant-name",
"PMUL0320-restaurant-restaurant-bookday", "PMUL0320-taxi-taxi-leaveat", "PMUL0320-taxi-taxi-destination"
"PMUL0320-taxi-taxi-departure", "PMUL0320-hotel-hotel-pricerange", "PMUL0320-hotel-hotel-type",
"PMUL0320-hotel-hotel-parking", "PMUL0320-hotel-hotel-bookday", "PMUL0320-hotel-hotel-bookpeople",
"PMUL0320-hotel-hotel-bookstay", "PMUL0320-hotel-hotel-internet", "PMUL0320-hotel-hotel-name",
"MUL2155-train-train-arriveby", "MUL2155-train-train-departure", "MUL2155-train-train-day",
"MUL2155-train-train-destination", "MUL2155-hotel-hotel-type", "MUL2155-hotel-hotel-bookday",
"MUL2155-hotel-hotel-bookpeople", "MUL2155-hotel-hotel-bookstay", "MUL2155-hotel-hotel-stars",
"MUL2155-hotel-hotel-internet", "MUL2155-hotel-hotel-name", "MUL2155-hotel-hotel-area",
"PMUL0815-restaurant-restaurant-pricerange", "PMUL0815-restaurant-restaurant-food",
"PMUL0815-restaurant-restaurant-name", "PMUL0815-train-train-departure", "PMUL0815-train-train-day",
"PMUL0815-train-train-bookpeople", "PMUL0815-train-train-leaveat", "PMUL0815-train-train-destination",
"PMUL3263-restaurant-restaurant-name", "PMUL3263-restaurant-restaurant-bookday",
"PMUL3263-restaurant-restaurant-bookpeople", "PMUL3263-restaurant-restaurant-booktime",
"PMUL3263-train-train-arriveby", "PMUL3263-train-train-departure", "PMUL3263-train-train-day",
"PMUL3263-train-train-bookpeople", "PMUL3263-train-train-destination", "SNG0423-train-train-arriveby",
"SNG0423-train-train-departure", "SNG0423-train-train-day", "SNG0423-train-train-bookpeople",
```

2. question answering

利用前段所產生之label，sentence 1 形式依舊為”{service} {slot}: {slot description}” ，sentence 2 為整段對話，送入BERT qa model當中，讓model去dialogue當中尋找相對應的slot value。將answer span 的開始結束位置取出，再將tokenize後的context index list取出，檢查answer index 在context index list中的位置，若是不同的話則更新answer index 的值。

模型部分，使用bert-base-uncased，tokenozer也使用bert-base-uncased。我們發現對話的最常長度為760個單字，因此將max length設置為760以確保不要錯過任何一個訊息，又不至於生成過多的zero factor浪費記憶體空間。利用QA model輸出的start/ end logit，將此兩參數加總輸出，取出相加分數最高的組合作為輸出。訓練集使用126566筆資料進行訓練，epoch=5，Gradient Accumulation steps=16，batch size=2，optimizer=Adam，learning rate=5e-5，vocab size=21128，hidden size=768，hidden layer=12。使用單張2070 ti顯示卡進行運算。訓練時間約5-10小時

B. NLG(自然對話生成):

Chit-chat模型訓練:

1. 資料訓練:

訓練資料為過去好的chit-chat候選資料，首先在input資料進Parlai之前，要先做格式整理，其中一筆資料形式如下:

- text:
  - user: Can you help me check the weather for March 1st for Campbell?
  - system: Sure. There will be a 4 percent change of rain with an average temperature of 91 degrees Fahrenheit.
  - user: OK. Can you find some Electropop songs? I'd like some Halsey.
- labels:
  - Halsey is a great artist.

模型為Tutorial Transformer Generator，其為一小型transformer(90M個參數)之預訓練模型

然後參考自parlai train model 的指令(https://parl.ai/docs/cli_usage.html#train-model)，輸入:

parlai train_model -t fromfile:parlaiformat --fromfile_datapath ./parlai --fromfile-datatype-extension true -m transformer/generator --init-model zoo:tutorial_transformer_generator/model --dict-file zoo:tutorial_transformer_generator/model.dict

超參數為:

--embedding-size 512

--n-layers 8

--ffn-size 2048

--dropout 0.1 (dropout = 0.1)

--n-heads 16

--learn-positional-embeddings True

--n-positions 512

--variant xlm

--activation gelu

--skip-generation True

--fp16 True

--text-truncate 512

--label-truncate 128

--dict-tokenizer bpe

--dict-lower True

-lr 1e-06 (learning rate = 10^-6)

--optimizer adamax (optimizer = adamax)

--lr-scheduler reduceonplateau

--gradient-clip 0.1

-veps 0.25

--betas 0.9,0.999

--update-freq 1

--attention-dropout 0.0

--relu-dropout 0.0

--skip-generation True

-vp 15

-stim 60

-vme 20000

-bs 16

-vmt ppl

-vmm min

2. 資料預測:

然後參考自parlai train model 的指令
(https://parl.ai/docs/cli_usage.html?fbclid=IwAR1b99y5nrDn2s9trSIidbq2QPPp3s3dy7NrkN2HIQD2cZY6yfFG0_Sd_Ws#interactive)

在parlai中預測的工具包稱為interactive

輸入: parlai interactive -mf ./train_90M < lm.input.test.cc.txt > lm.output.test.cc.txt

資料形式為一次輸入三筆user與system之間交互之對話紀錄且以遞增形式呈現，而經過interactive預測後會產生對應輸入三筆接連發生的回應，表示如下:

輸入:

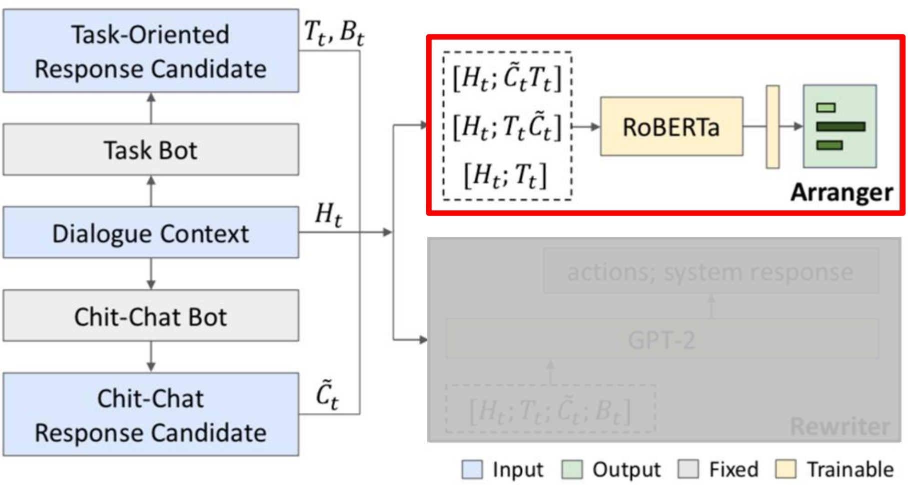
- user: I need some bus tickets. '[DONE]'
- user: I need some bus tickets. system: Sure, i can help you. Where are you departing from? user: San Diego on the 8th of this month. '[DONE]'
- user: I need some bus tickets. system: Sure, i can help you. Where are you departing from? user: San Diego on the 8th of this month. system: ok, how many tickets and where are you traveling too? What time would you like to depart? user: I need one ticket to Anaheim and if I could leave at 4:15 pm would be great. '[DONE]'

輸出:

- you should check it out .
- that ' s a good price .
- thank you .

1. Arranger評估：

概念圖(紅框處):



該模型在不被干預的情況下安排了現成的對話模型和現成的chit-chat模型的輸出。該模型將 (i) 對話歷史 Ht，(ii) 基於 Ht 的chit-chat模型生成的chit-chat response Ct 作為輸入，以及 (iii)基於Ht的面向任務的對話模型生成的task-oriented的response Tt。而模型輸入為Ht及Ct Tt， Ht及Tt Ct ，還有Ht及獨立的Tt具體來說，該模型通過 RoBERTa 對 Ht 和這三個response中的每一個的串聯進行編碼，並將得到的表示通過線性加 softmax 層進行選擇。為了訓練模型，我們通過將 ACCENTOR-SGD 訓練集中的turn t 的每個chit-chat候選者視為 Ct 並將ground-truth task-oriented response 視為 Tt 並基於以下條件

設置目標選擇來形成訓練實例候選的label（即好/壞）和位置（即response的開始/結束）。

利用ACCENTOR中的arrenger工具包針對parlai interactive output data的chit chat預測出三個機率值，分別為0, 1, 2，0: bad、1: good, beginning、2: good, end。模型為RoBERTa，重要超參數:learning rate=2*10^-5, epoch=3, batch size=24, max length=512，以下展示輸入資料及輸出資料:

輸入:

- context:
  - user: I need some bus tickets. system: Sure, i can help you. Where are you departing from? user: San Diego on the 8th of this month. system: ok, how many tickets and where are you traveling too? What time would you like to depart? user: I need one ticket to Anaheim and if I could leave at 4:15 pm would be great. system: Sure, please confirm leaving Anaheim on March 8th at 4:15 pm for 1 economy ticket for San Diego departure. Is this right? user: Yes, What station would I arrive at? What station am I leaving from? system: I apologize I am unable to book your request. Can we try an different request? user: could you look up tickets to Vegas departing at 5 pm from Fresno? system: Sure, please confirm: 1 economy ticket for Las Vegas departing Fresno at 5 pm on March 8th. Is that right? user: yes it is. system: I apologize i am unable to book that ticket, may I suggest the 5:30 pm bus, for $43? user: Yes, please book it.
- gt:
  - Your bus ticket is confirmed.
- chit chat:
  - Thank you
- labels: 0

輸出:

```
chit chat: you should check it out .
weights: [0.30349916, -2.9090686, -5.8177714]
------
chit chat: that ' s a good price .
weights: [0.6920911, -0.55344236, -6.267423]
------
chit chat: thank you .
weights: [0.17695992, -1.5541675, -6.287538]
------
chit chat: i can recommend some places .
weights: [-2.2216084, -4.5952363, -6.4438972]
------
chit chat: it is a great place to eat .
weights: [0.0008546198, -1.5127692, -6.137079]
------
```

由上圖所示，據輸出之權重之最大值去判定是0、1或是2，如第一筆[0.30349916, -2.9090686, -5.8177714]中最大的值為0.30349916，就判定為0，是一不好的chit-chat。全部預測資料中，判斷為0: bad、1: good, beginning、2: good, end的個數為，0:10795

個、1:3922個、2:1561個。最後將資料整理為作業要求的形式如下:

整理:

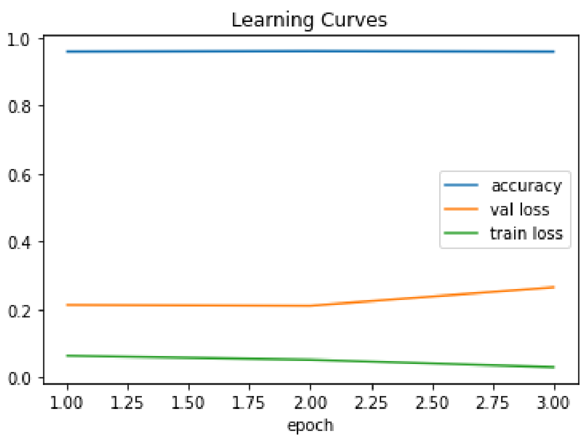
```
'PMUL0320': {1: {'start': '', 'end': '', 'mod': ''},
 3: {'start': '', 'end': '', 'mod': ''},
 5: {'start': '', 'end': '', 'mod': ''},
 7: {'start': '', 'end': '', 'mod': ''},
 9: {'start': '', 'end': '', 'mod': ''},
 11: {'start': '', 'end': '', 'mod': ''},
 13: {'start': '', 'end': 'what do you think ?', 'mod': ''},
 15: {'start': '', 'end': '', 'mod': ''},
 17: {'start': '', 'end': '', 'mod': ''},
 19: {'start': '', 'end': '', 'mod': ''},
 21: {'start': 'you are very welcome .', 'end': '', 'mod': ''}},
'MUL2155': {1: {'start': '', 'end': '', 'mod': ''},
 3: {'start': '', 'end': '', 'mod': ''},
 5: {'start': '', 'end': 'do you have any other questions ?', 'mod': ''},
 7: {'start': '', 'end': '', 'mod': ''},
 9: {'start': '', 'end': '', 'mod': ''},
 11: {'start': '',
  'end': 'do you think that would be a good place to stay ?',
  'mod': ''},
```

上圖為最後繳交給助教之格式，呈現不同對話群組中chit-chat及位置資訊。

3. 結論

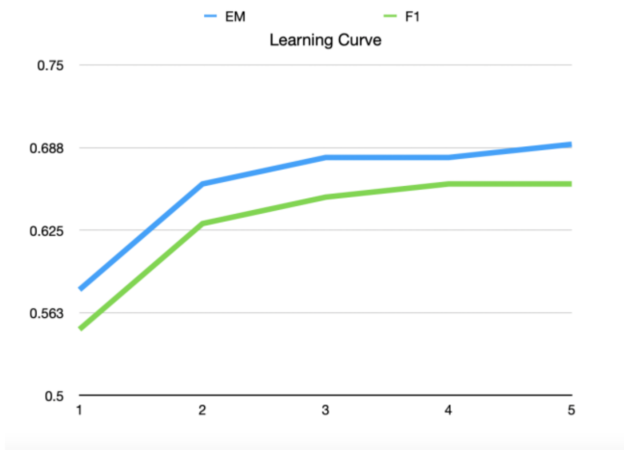
A. DST:

1. text classification:



accuracy部分採用total accuracy，即為: (正確預測的slot)/(所有的slot)， 此部分準確率為95~96%。val loss從0.212些微上升到0.264，train loss則由0.063下降到0.029。

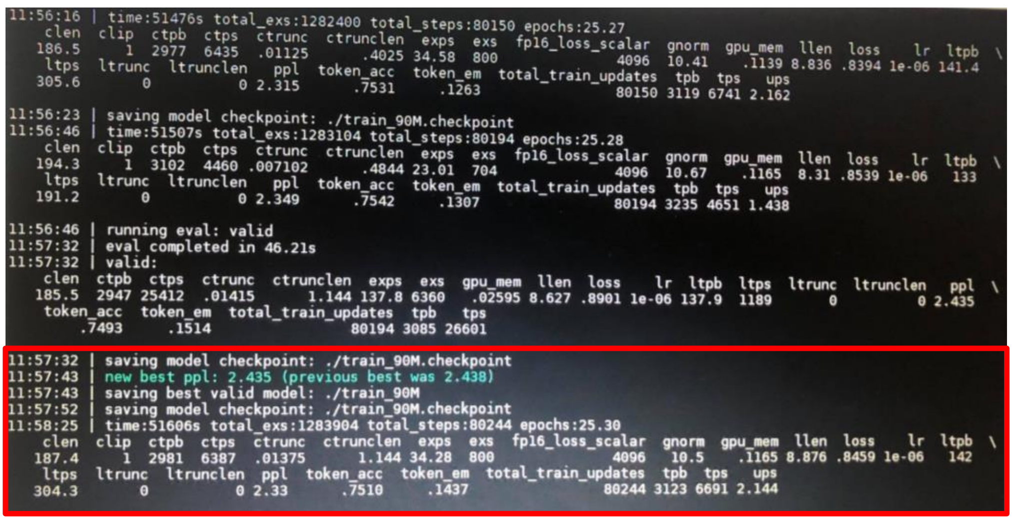
2.question answering



評分的方式使用exact match 和 f1 score。可以看到隨著epoch增加，EM和F1都隨之增加，在第三個epoch後開始飽和，最終EM=0.691，F1=0.652

B. NLG:

Chit-chat:



圖中顯示最後一個epoch跑出的結果，token accuracy到了0.7510

Arranger:

最終RoBERTa模型的結果為:

train_acc = 0.7189145620634068

train_loss = 0.5734808349732271

eval_acc = 0.6631650079862391

eval_loss = 0.6175413059253319

4. 工作分配

DST: 王志中 吳承哲

NLG: 莊詠竣 賴怡穎 吳承哲

報告撰寫: 王志中 吳承哲 莊詠竣 賴怡穎

口頭報告: 莊詠竣、吳承哲

投影片: 賴怡穎、吳承哲