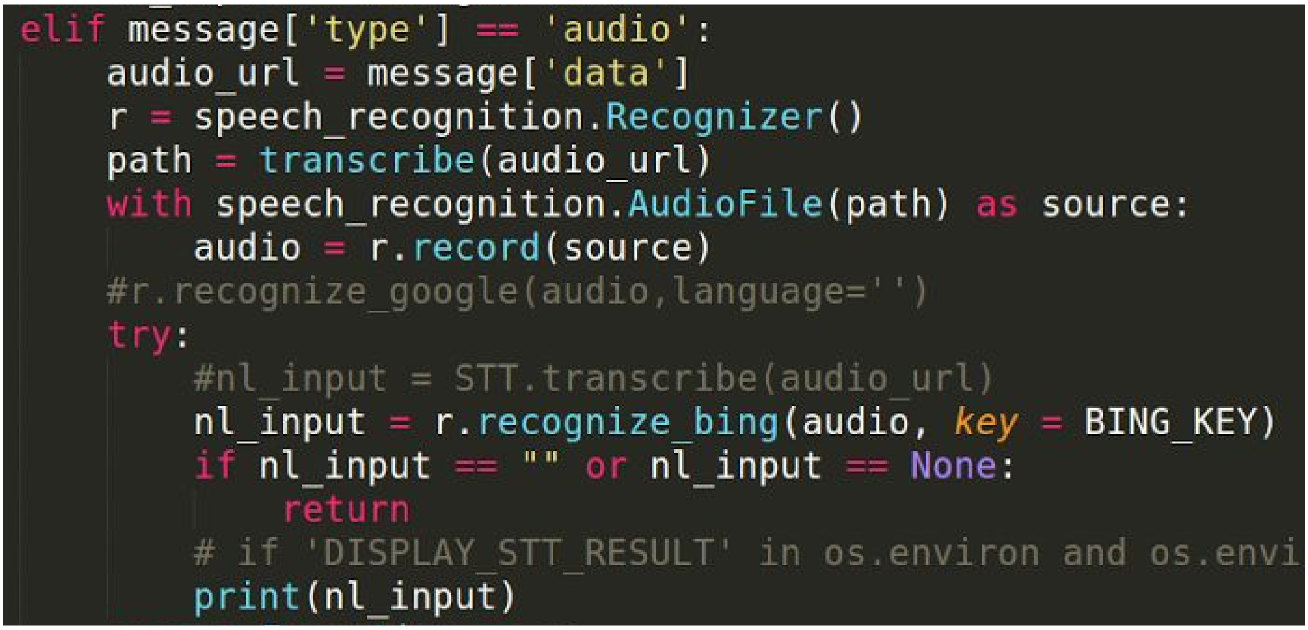
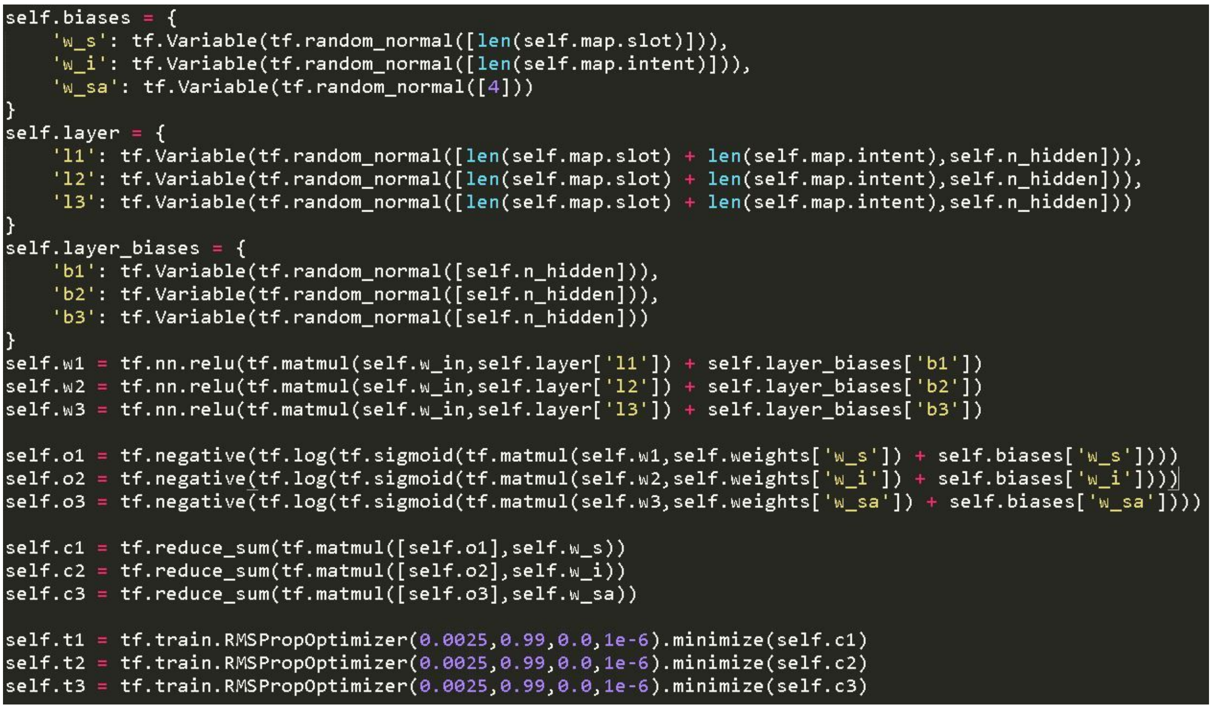
1. Speech multimodal api
   我們用的是python中一個叫作SpeechRecognition的package,他可以接google的api、ibm的、microsoft bing的、因為之前申請了azure的帳號，所以我們最後使用microsoft的api，主要就是先判斷messenger傳過來的訊息是否為audio形式，如果是的話，就把他下載下來轉成wav檔，透過SpeechRecognition把他轉成NL在當成input餵進去model裏面這樣。
   程式如下：
   先把speech_recognition叫出來，接下來把facebook傳過來的語音url轉成wav檔(透過transcribe還有speech_recognition吃wav檔)，之後在把他轉成他可以吃的形式(speech_recognition.AudioFile)最後透過recognize_bing(微軟azure api)去進行語音辨識得到我們的nl_input。



```
elif message['type'] == 'audio':
    audio_url = message['data']
    r = speech_recognition.Recognizer()
    path = transcribe(audio_url)
    with speech_recognition.AudioFile(path) as source:
        audio = r.record(source)
    #r.recognize_google(audio,language='')
    try:
        #nl_input = STT.transcribe(audio_url)
        nl_input = r.recognize_bing(audio, key = BING_KEY)
        if nl_input == "" or nl_input == None:
            return
        # if 'DISPLAY_STT_RESULT' in os.environ and os.envi
        print(nl_input)
```

2. Reinforcement learning based dialogue policy:
   Environment State的設定包含使用者曾經講過的slot & intent tag,與System所做的history action及user simulator對於system的行為所給予的reward,根據User & System曾經的互動,學習應該做的Dialogue Policy.
   這次為了做reinforcement learning,User simulator做了更細微的reward 更動,如果system confirm一些跟當前對話無關的intent or slot tags，或是再次詢問了user simulator已經給過的slot tag，我們會給負的reward,如果system confirm到正確的intent or slot就會給予

獎勵.希望藉由更詳細的更動讓reinforcement learning 學得更快.
前面會先跑5000次對話,全部都是random的explore,但會有基本的邏輯(不會以A intent配B intent的slot作出詢問, ex: What currency whould you like to know about this stock information. intent: query, slot: country (get_exchange_rate的slot)),再來random的機率會線性下降,讓model慢慢掌握抉擇system action的工作,超過10000次之後就完全是model在做選擇。

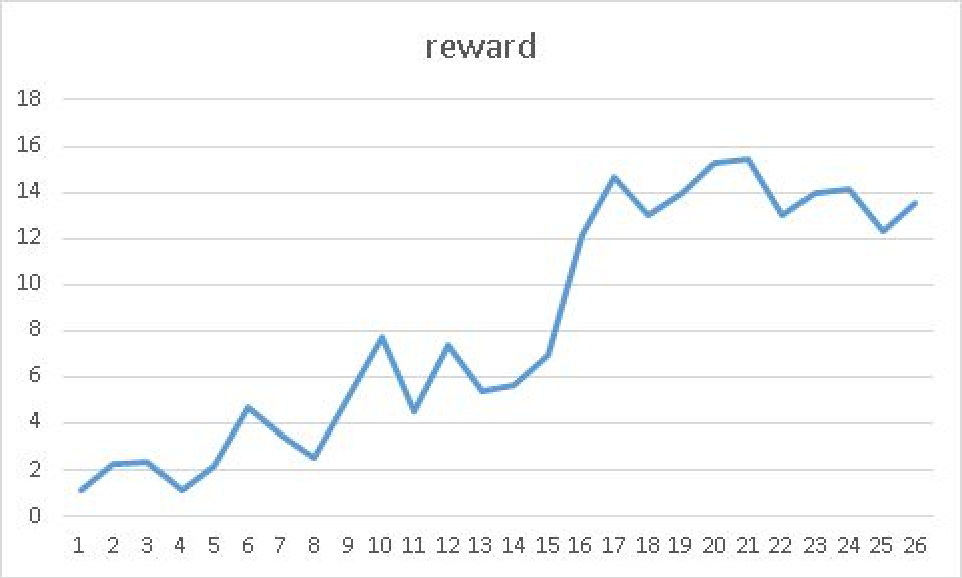


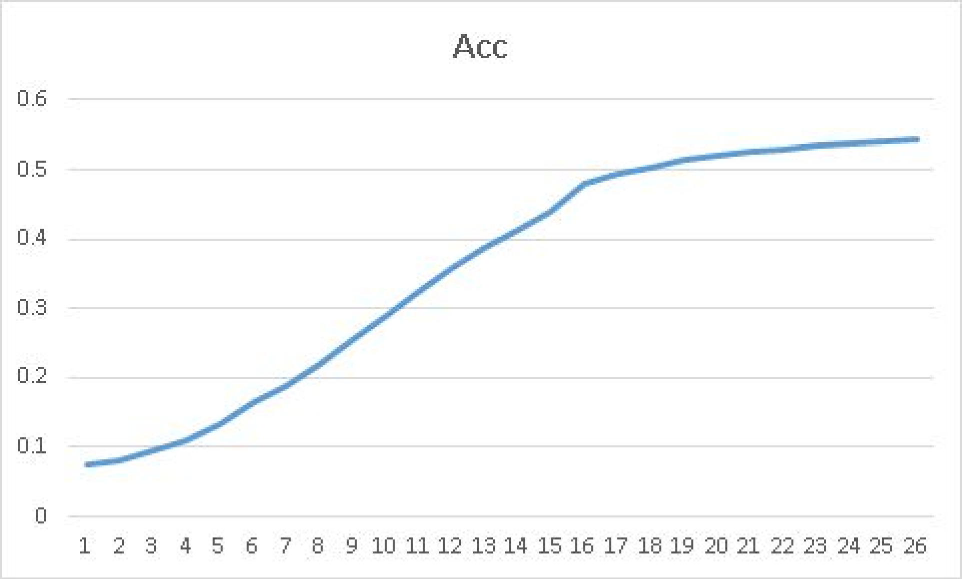
```
self.biases = {
    'w_s': tf.Variable(tf.random_normal([len(self.map.slot)])),
    'w_i': tf.Variable(tf.random_normal([len(self.map.intent)])),
    'w_sa': tf.Variable(tf.random_normal([4]))
}
self.layer = {
    'l1': tf.Variable(tf.random_normal([len(self.map.slot) + len(self.map.intent),self.n_hidden])),
    'l2': tf.Variable(tf.random_normal([len(self.map.slot) + len(self.map.intent),self.n_hidden])),
    'l3': tf.Variable(tf.random_normal([len(self.map.slot) + len(self.map.intent),self.n_hidden]))
}
self.layer_biases = {
    'b1': tf.Variable(tf.random_normal([self.n_hidden])),
    'b2': tf.Variable(tf.random_normal([self.n_hidden])),
    'b3': tf.Variable(tf.random_normal([self.n_hidden]))
}
self.w1 = tf.nn.relu(tf.matmul(self.w_in,self.layer['l1']) + self.layer_biases['b1'])
self.w2 = tf.nn.relu(tf.matmul(self.w_in,self.layer['l2']) + self.layer_biases['b2'])
self.w3 = tf.nn.relu(tf.matmul(self.w_in,self.layer['l3']) + self.layer_biases['b3'])

self.o1 = tf.negative(tf.log(tf.sigmoid(tf.matmul(self.w1,self.weights['w_s']) + self.biases['w_s'])))
self.o2 = tf.negative(tf.log(tf.sigmoid(tf.matmul(self.w2,self.weights['w_i']) + self.biases['w_i'])))
self.o3 = tf.negative(tf.log(tf.sigmoid(tf.matmul(self.w3,self.weights['w_sa']) + self.biases['w_sa'])))

self.c1 = tf.reduce_sum(tf.matmul([self.o1],self.w_s))
self.c2 = tf.reduce_sum(tf.matmul([self.o2],self.w_i))
self.c3 = tf.reduce_sum(tf.matmul([self.o3],self.w_sa))

self.t1 = tf.train.RMSPropOptimizer(0.0025,0.99,0.0,1e-6).minimize(self.c1)
self.t2 = tf.train.RMSPropOptimizer(0.0025,0.99,0.0,1e-6).minimize(self.c2)
self.t3 = tf.train.RMSPropOptimizer(0.0025,0.99,0.0,1e-6).minimize(self.c3)
```

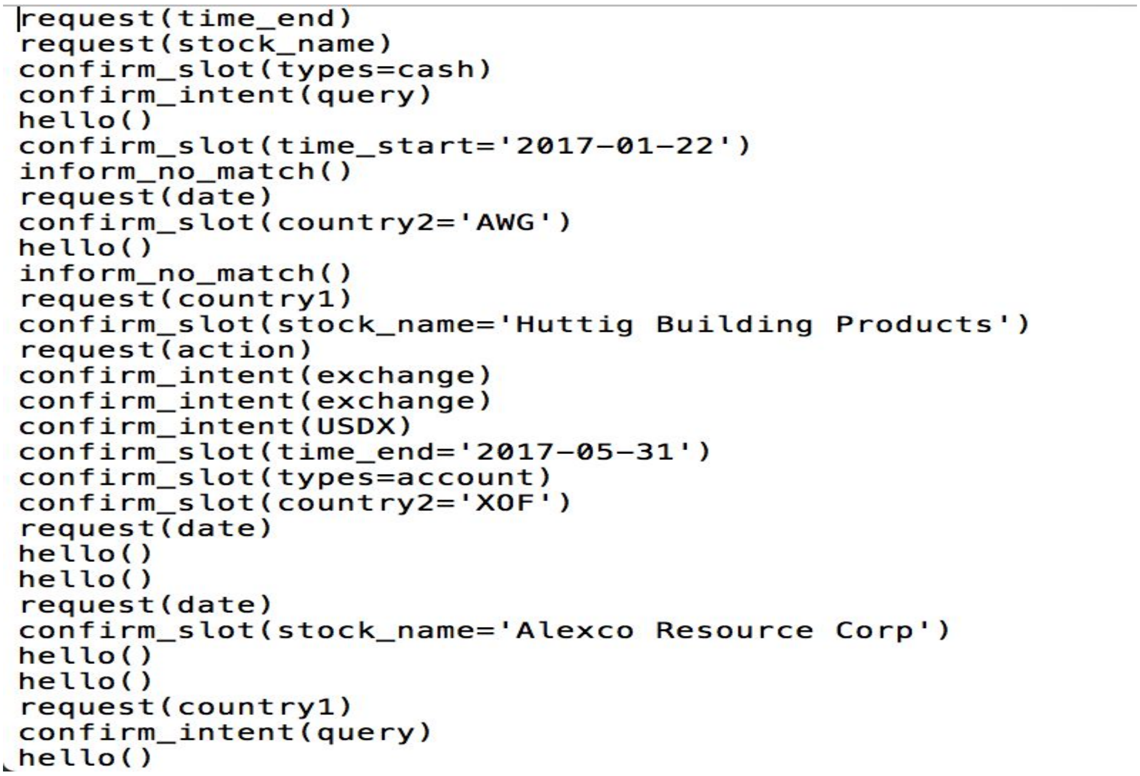




3. NN-based NLG

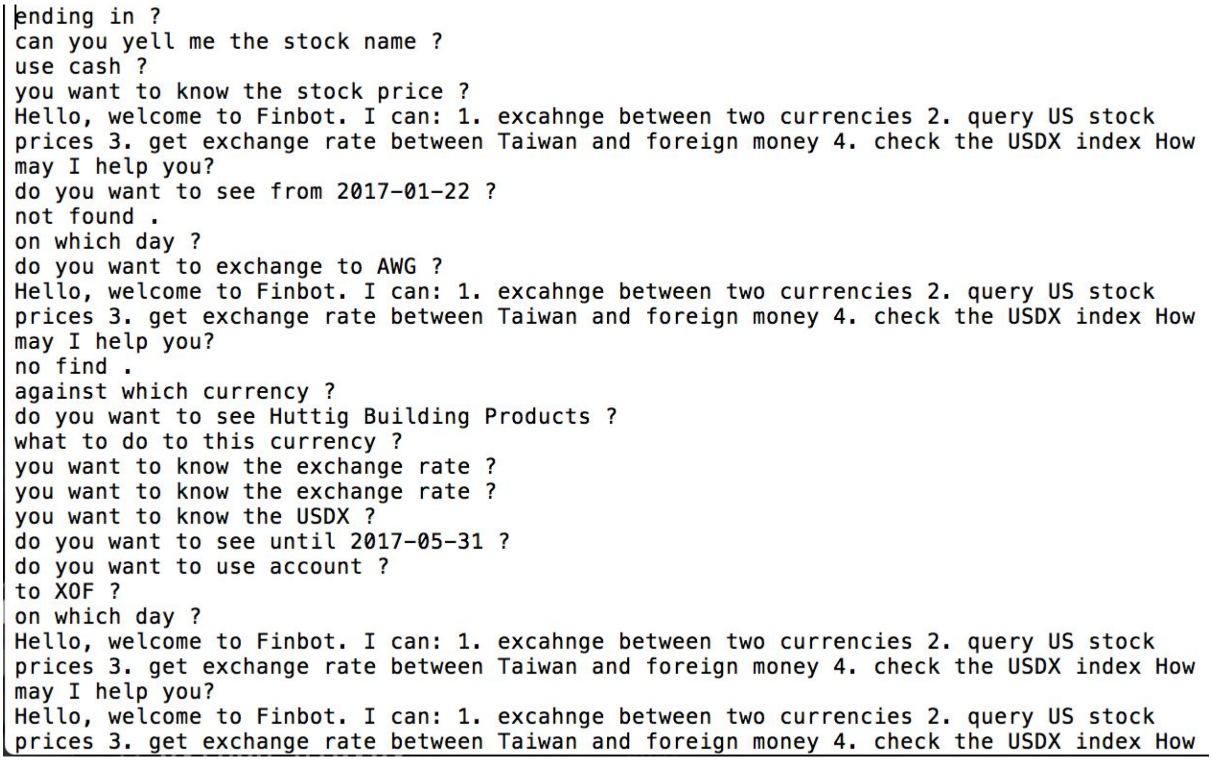
- implementation:
  我們的 model 參考自 tgen, A statistical natural language generator for spoken dialogue systems(連結)架構上使用 sequence to sequence recurrent neural network 並以 tensorflow 實作
- training data
  dialogue acts: 包含hello(), inform_no_match(), confirm_intent([slot]), confirm_slot([slot]), request([slot]), 及 response([slot]) 配上相對應的slots

  dialogue acts



```
request(time_end)
request(stock_name)
confirm_slot(types=cash)
confirm_intent(query)
hello()
confirm_slot(time_start='2017-01-22')
inform_no_match()
request(date)
confirm_slot(country2='AWG')
hello()
inform_no_match()
request(country1)
confirm_slot(stock_name='Huttig Building Products')
request(action)
confirm_intent(exchange)
confirm_intent(exchange)
confirm_intent(USDX)
confirm_slot(time_end='2017-05-31')
confirm_slot(types=account)
confirm_slot(country2='XOF')
request(date)
hello()
hello()
request(date)
confirm_slot(stock_name='Alexco Resource Corp')
hello()
hello()
request(country1)
confirm_intent(query)
hello()
```

text:



```
ending in ?
can you yell me the stock name ?
use cash ?
you want to know the stock price ?
Hello, welcome to Finbot. I can: 1. excahnge between two currencies 2. query US stock
prices 3. get exchange rate between Taiwan and foreign money 4. check the USDX index How
may I help you?
do you want to see from 2017-01-22 ?
not found .
on which day ?
do you want to exchange to AWG ?
Hello, welcome to Finbot. I can: 1. excahnge between two currencies 2. query US stock
prices 3. get exchange rate between Taiwan and foreign money 4. check the USDX index How
may I help you?
no find .
against which currency ?
do you want to see Huttig Building Products ?
what to do to this currency ?
you want to know the exchange rate ?
you want to know the exchange rate ?
you want to know the USDX ?
do you want to see until 2017-05-31 ?
do you want to use account ?
to XOF ?
on which day ?
Hello, welcome to Finbot. I can: 1. excahnge between two currencies 2. query US stock
prices 3. get exchange rate between Taiwan and foreign money 4. check the USDX index How
may I help you?
Hello, welcome to Finbot. I can: 1. excahnge between two currencies 2. query US stock
prices 3. get exchange rate between Taiwan and foreign money 4. check the USDX index How
```

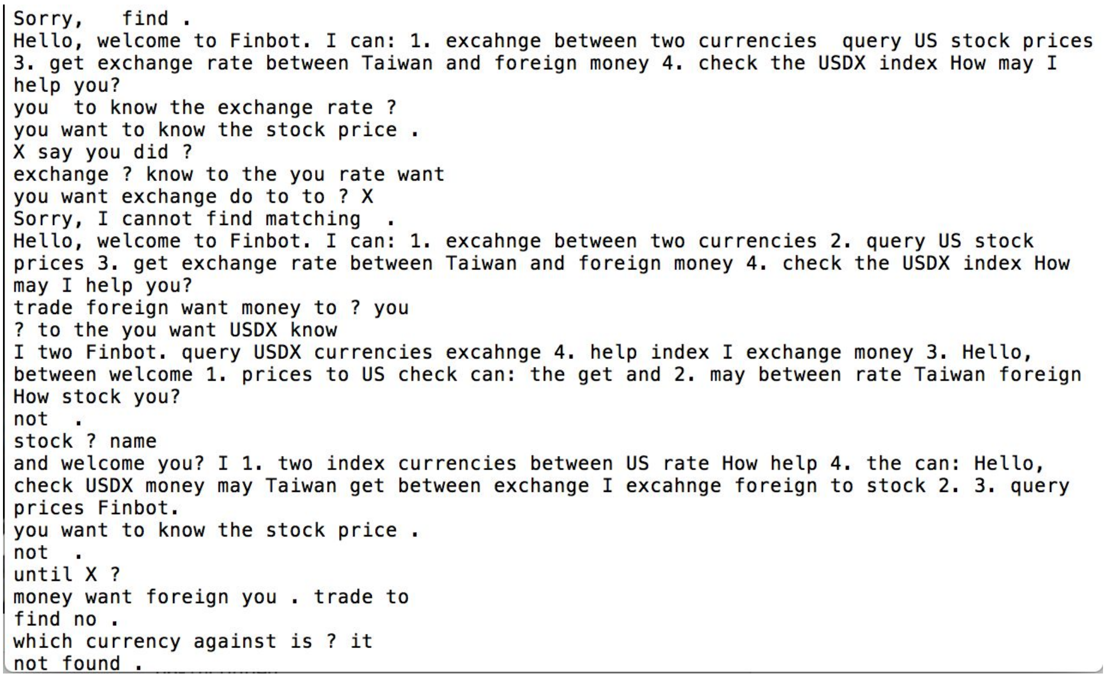
- training/testing split vs blue score:

  90% for training 0.310810689481

  80% for training 0.205758816209

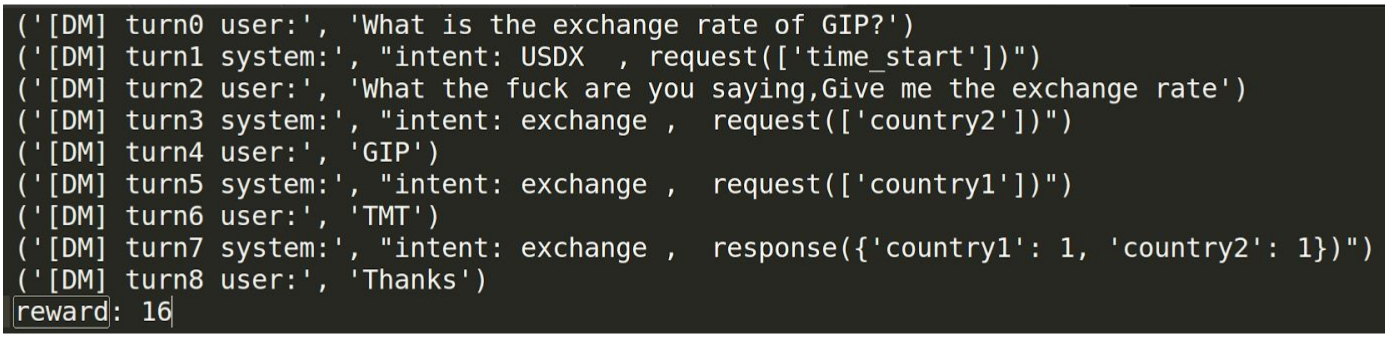
  70% for training 0.0926496039644

- testing result:

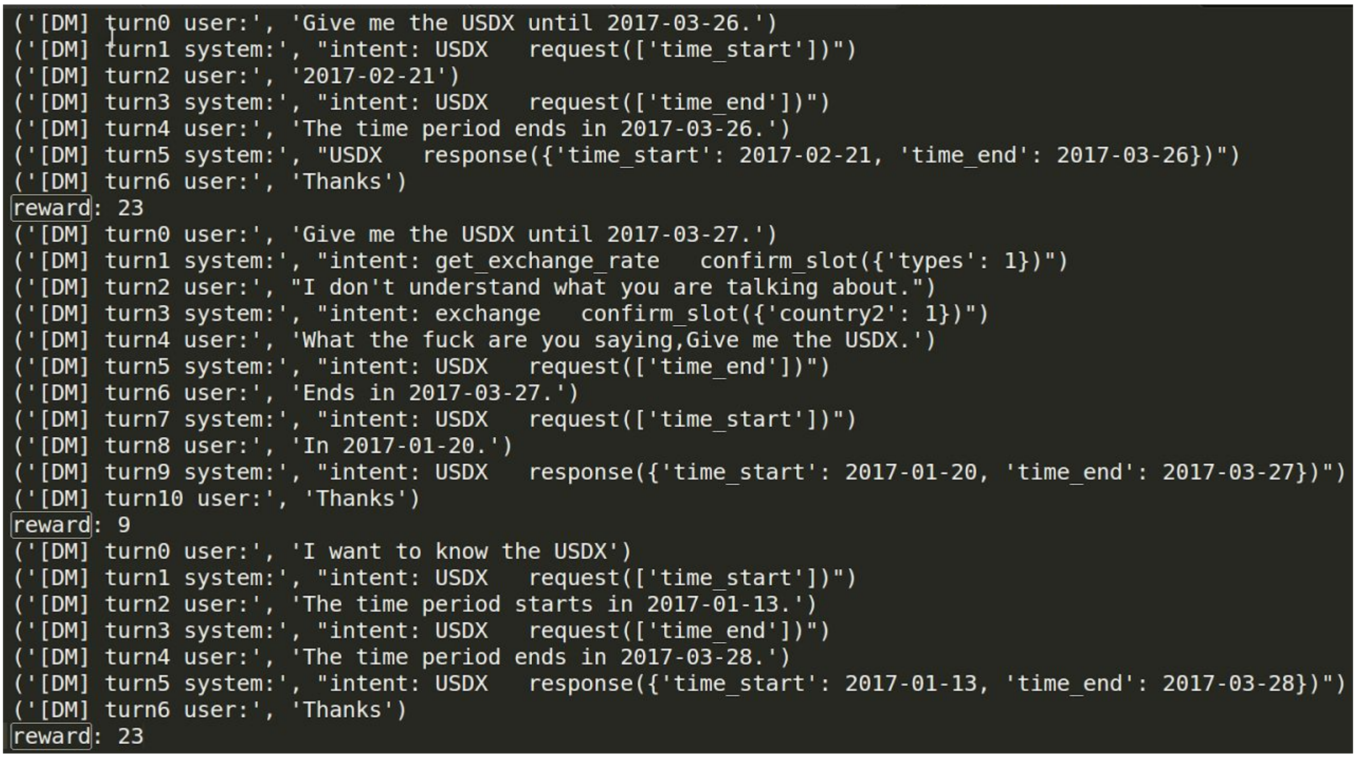


```
Sorry,   find .
Hello, welcome to Finbot. I can: 1. excahnge between two currencies  query US stock prices
3. get exchange rate between Taiwan and foreign money 4. check the USDX index How may I
help you?
you  to know the exchange rate ?
you want to know the stock price .
X say you did ?
exchange ? know to the you rate want
you want exchange do to to ? X
Sorry, I cannot find matching  .
Hello, welcome to Finbot. I can: 1. excahnge between two currencies 2. query US stock
prices 3. get exchange rate between Taiwan and foreign money 4. check the USDX index How
may I help you?
trade foreign want money to ? you
? to the you want USDX know
I two Finbot. query USDX currencies excahnge 4. help index I exchange money 3. Hello,
between welcome 1. prices to US check can: the get and 2. may between rate Taiwan foreign
How stock you?
not  .
stock ? name
and welcome you? I 1. two index currencies between US rate How help 4. the can: Hello,
check USDX money may Taiwan get between exchange I excahnge foreign to stock 2. 3. query
prices Finbot.
you want to know the stock price .
not  .
until X ?
money want foreign you . trade to
find no .
which currency against is ? it
not found .
```

4. Performance for simulated dialogues：



```
('[DM] turn0 user:', 'What is the exchange rate of GIP?')
('[DM] turn1 system:', "intent: USDX  , request(['time_start'])")
('[DM] turn2 user:', 'What the fuck are you saying,Give me the exchange rate')
('[DM] turn3 system:', "intent: exchange ,  request(['country2'])")
('[DM] turn4 user:', 'GIP')
('[DM] turn5 system:', "intent: exchange ,  request(['country1'])")
('[DM] turn6 user:', 'TMT')
('[DM] turn7 system:', "intent: exchange ,  response({'country1': 1, 'country2': 1})")
('[DM] turn8 user:', 'Thanks')
reward: 16
```



```
('[DM] turn0 user:', 'Give me the USDX until 2017-03-26.')
('[DM] turn1 system:', "intent: USDX   request(['time_start'])")
('[DM] turn2 user:', '2017-02-21')
('[DM] turn3 system:', "intent: USDX   request(['time_end'])")
('[DM] turn4 user:', 'The time period ends in 2017-03-26.')
('[DM] turn5 system:', "USDX   response({'time_start': 2017-02-21, 'time_end': 2017-03-26})")
('[DM] turn6 user:', 'Thanks')
reward: 23
('[DM] turn0 user:', 'Give me the USDX until 2017-03-27.')
('[DM] turn1 system:', "intent: get_exchange_rate   confirm_slot({'types': 1})")
('[DM] turn2 user:', "I don't understand what you are talking about.")
('[DM] turn3 system:', "intent: exchange   confirm_slot({'country2': 1})")
('[DM] turn4 user:', 'What the fuck are you saying,Give me the USDX.')
('[DM] turn5 system:', "intent: USDX   request(['time_end'])")
('[DM] turn6 user:', 'Ends in 2017-03-27.')
('[DM] turn7 system:', "intent: USDX   request(['time_start'])")
('[DM] turn8 user:', 'In 2017-01-20.')
('[DM] turn9 system:', "intent: USDX   response({'time_start': 2017-01-20, 'time_end': 2017-03-27})")
('[DM] turn10 user:', 'Thanks')
reward: 9
('[DM] turn0 user:', 'I want to know the USDX')
('[DM] turn1 system:', "intent: USDX   request(['time_start'])")
('[DM] turn2 user:', 'The time period starts in 2017-01-13.')
('[DM] turn3 system:', "intent: USDX   request(['time_end'])")
('[DM] turn4 user:', 'The time period ends in 2017-03-28.')
('[DM] turn5 system:', "intent: USDX   response({'time_start': 2017-01-13, 'time_end': 2017-03-28})")
('[DM] turn6 user:', 'Thanks')
reward: 23
```

avg_reward: 13.5723

succ_rate: 0.5113

avg_turns: 6.7894

avg_turnacc:0.87292420078228117

用 user simulator跑了大概10000筆資料結果如上，跟rl train出來的結果效果是略低，但還在誤差值裡面。

Facebook Bot:https://www.facebook.com/Finbot-1892124104332642/

因為還沒通過上市申請,如果助教要測請給我們facebook名子,加入測試人員account, Bot就會回了.