

# MusicBot Milestone 3

Members : 王耀賢 陳品融 陳聖曄 李俊鈺 謝明勳

## Speech API

使用Microsoft Azure的Bing Speech API來進行語音到文字的轉換。Bing Speech API不支援自動偵測語言，需要先設定好是要進行哪一種語言的辨識。而目前在我們的網頁界面當中，語音輸入的部份預設的語言為中文。

## RL Based Dialogue Policy

### 1. Method:

我們使用Policy gradient來做RL，在training時是讓DM跟user simulator自由去互動，並且把互動的data放到memory中，再定期由memory中sample data來更新policy Network。

user的input sentence進來後會先用rule-based做dialogue state tracking產生一個current dialogue state，dialogue state再變成一個vector，餵給policy network產生一個action distribution，最後根據這個action distribution產生一個action，並且用這個action產生action frame以及natural language response。

由於RL一開始的policy很爛，幾乎拿不到reward，因此我們有先用rule-based的policy sample出一些example加進experience\_replay\_pool裡。

### 2. State vector:

我們必須把原本的state轉成vector的形式才能作為RL model的input，因此我們考慮了self.state以及self.confirmed\_state，最終轉成一個10維的vector。

self.state的形式為{'intent':{'search':p1, 'recommend':p2, 'info':p3}, 'slot':{'track':{}, 'artist':{}, 'genre':{}}，是NLU去抓intent及slot的結果；self.confirmed\_state則是已經向user confirm過的intent及slot，其形式為{'intent': None, 'slot':{'artist': None, 'track': None, 'genre': None}}。

首先self.state intent probs p1, p2, p3直接放進vector的前三維；接著考慮self.state的slot probs，如果self.state['slot']裡的有抓到東西且還沒confirm過的話，就把抓到的機率加到對應的維度裡，如果還沒抓到東西或者已被confirm過的話則該項為0，總共是3維；最後是考量self.confirmed\_state的intent及slot，分別是一個一維及三維的one-hot vector，如果對應到的intent或者slot已經confirm過的話，則令該項為1。所以這樣最後總共會是一個10維的vector。

### 3. Action:

['question\_intent', 'question\_slot\_track', 'question\_slot\_artist', 'question\_slot\_genre', 'confirm\_intent', 'confirm\_slot\_track', 'confirm\_slot\_artist', 'confirm\_slot\_genre', 'Response', 'info']

共有10種不同的action，而‘Response’，‘info’代表結束的action。

4. Model:

簡單的DNN，input是state vector，接一層size為50的hidden layer，而output size為num\_action(10)，過一個softmax的activation function得到action distribution

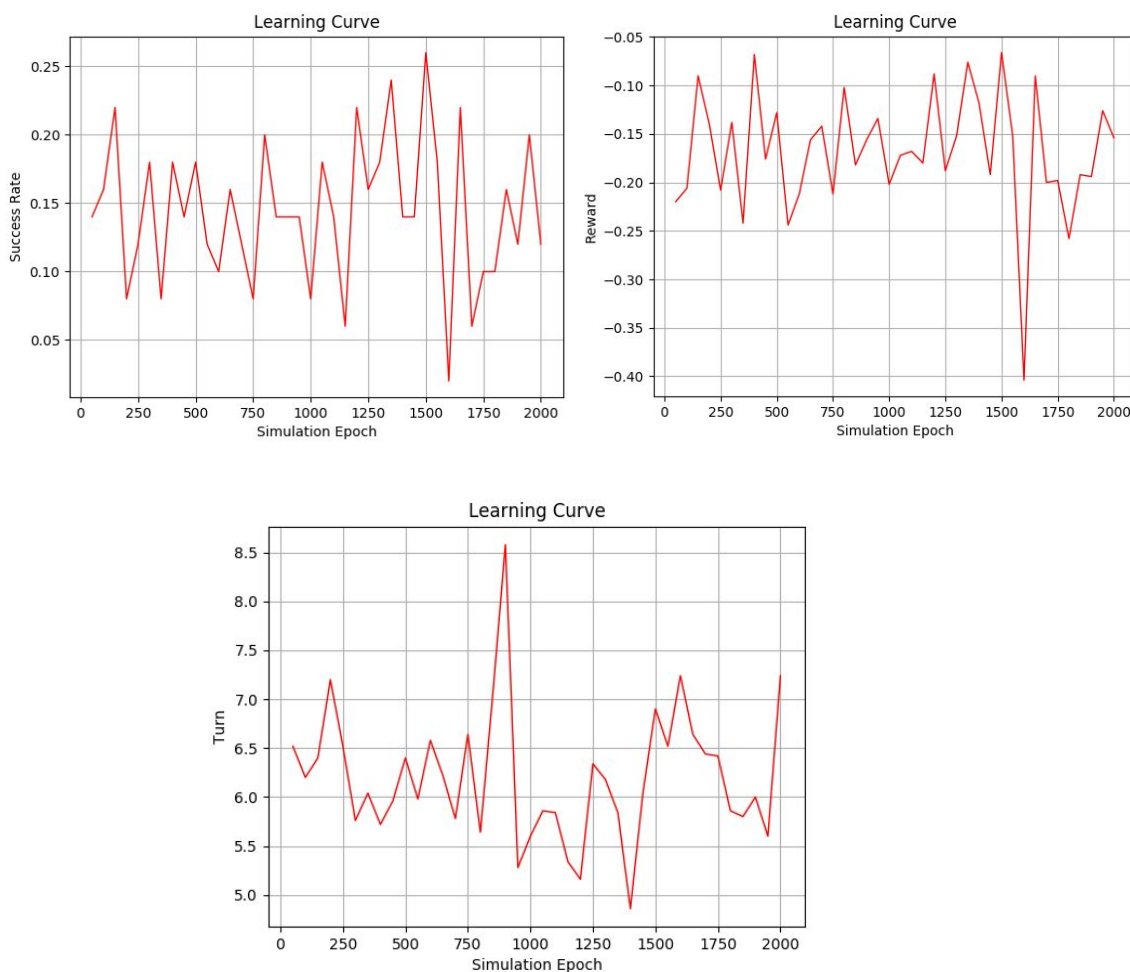
5. Model loss function:

$$loss = -Reward \times \log(p)$$

其中p是根據DNN output的action distribution去sample得到的

6. Performance:

表現沒有很好，simulation epoch對應success rate, reward, turns的結果如下圖(每50步測一次，每一次測50筆取平均)。



7. 分析：我們認為我們的rl agent performance不佳的原因有以下幾點：

1. 雖然已經有用rule-based的policy填了一些data到replay\_pool裡，但可能數量不夠，導致之後rl policy產生的一堆爛example進來後，sample出來的還是那些拿不到reward，無法有效update的data。

2. 由於我們的model爲一層hidden layer, size=50的DNN, 所以有可能過於簡單, 導致無法學會適當的policy。
3. Vector representation可能可以考慮更多factor, 如action history等等。

## NN-Based NLG

我們NN-based的NLG使用的是sequence to sequence的架構。給定一個 action 的 semantic frame, 這個 model 需要夠生成出一段句子。為了簡化training的過程並且減少noise的產生, input的action frame當中的slot只會以True或False的binary方式來表示, 而target sentence中的slot則是以slot所屬類別來表示。因此在我們的model生出了句子之後, 需要再將slot當中實際的質填入此句子當中。

### Input Action Vector

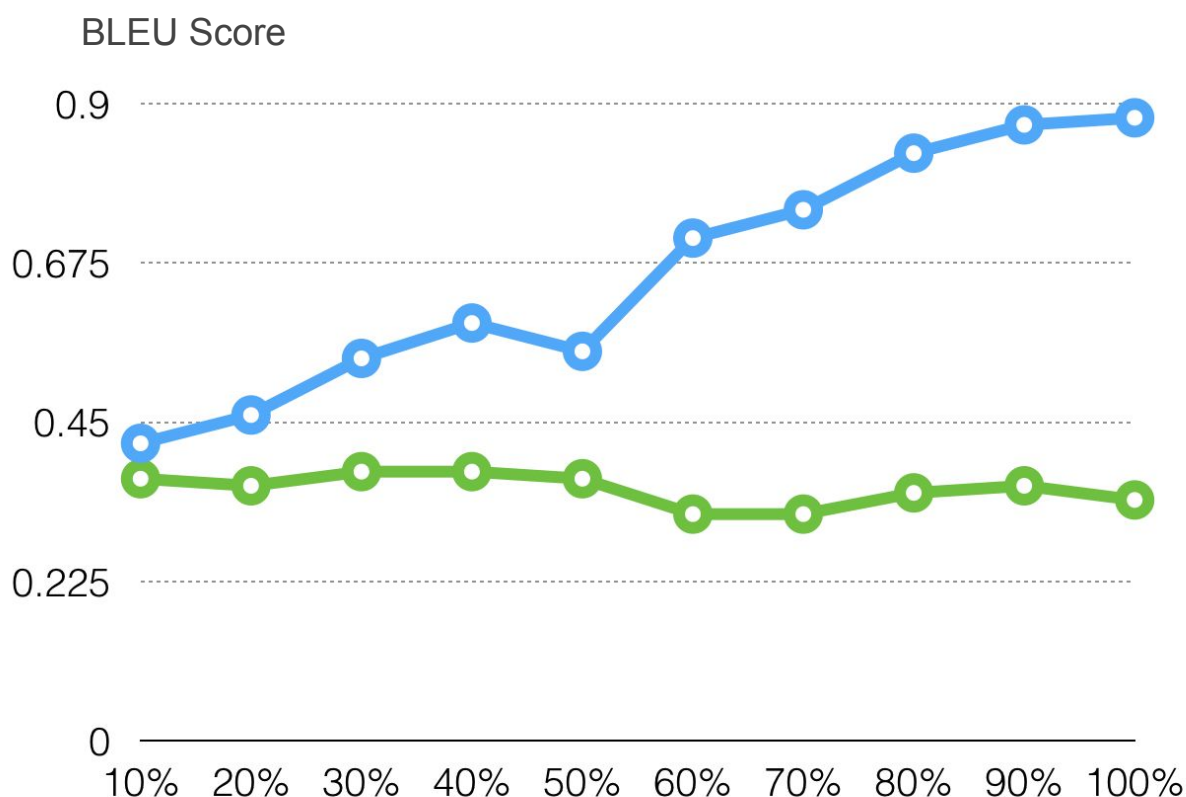
Input 的 vector 可以大致分為三個區塊, 分別為action的種類、intent的類別、以及各個slot是否有在此action當中。 action 的部份, 我們的action總共有四種, 分別為 question, confirm, response, info。而 intent 則有3種, search, recommend, info。以及3種slot, artist, track, genre。

### Training/Testing Data

我們人工生出了所有可能的action並為其做出了句子, 並將training和testing的部份分開來。

## Test Results

```
你好，請問需要什麼服務？
> question none True none none 0
請問你是否要填入歌手名稱？
> question none none True none 0
請問你是否要填入歌曲名稱？
> question none True True none 0
想聽哪個歌手的什麼歌？
> question none none none True 0
請問你是否要填入曲風？
> confirm search none none none 0
再確認一次，請問你是想搜尋嗎？
> confirm recommend none none none 0
再確認一次，請問你是想推薦嗎？
> confirm info none none none 0
再確認一次，請問你是想查相關資
> confirm none True none none 0
再確認一次，請問你是否曾填入歌
> confirm search True none none 0
你認一下，你是不是想要聽 a 的歌
> confirm recommend True none none 0
確認一下，你要我推薦 a 的歌嗎？
> confirm info True none none 0
你認一下，你是不是要知道 a 的歌
> confirm none none True none 0
再確認一次，請問你是否曾填入歌
> confirm none True True none 0
再確認一次，是選 a 的 t 嗎？
> confirm search none True none 0
你是想要聽 t 嗎？
> confirm search True True none 0
你是想要聽 a 的 t 嗎？
```



藍色為training, 綠色為testing。縱軸為BLEU score, 橫軸為使用的data 比例。

## Performance

對話範例(與RL agent對話成功的例子):

(Goal: intent:recommend slot\_track:好朋友的祝福 slot\_artist:A-lin)

User simulator:請推薦跟好朋友的祝福類似的歌曲

DM: 請問你想聽好朋友的祝福這個歌手的歌曲嗎?

User simulator: 不對 有關好朋友的祝福的歌曲

DM:請問你想要填入歌手嗎?

User Simulator: 有沒有跟A-Lin類似歌手的歌曲呢

DM:請問你想要填入歌手嗎?

User Simulator: 推薦A-lin風格

DM: 你是想聽歌嗎?

User Simulator: 不對 有關好朋友的祝福的歌曲

DM: 再確認一次 請問你是想推薦歌曲嗎?

User Simulator: 對啊

DM: 為你推薦 梁靜茹的別再為他流淚 丁噹的我沒那麼愛你 莊心妍的媽媽你去哪兒 吳克群的不會痛的不叫愛情

Demonstration (5%)

<http://140.112.90.203:8888>