PPT BOT2
Milestone3

r04921006 謝仲凱 r05942039 許宏瑋
r05942046 游肇輝 r05921066 鄭培淳

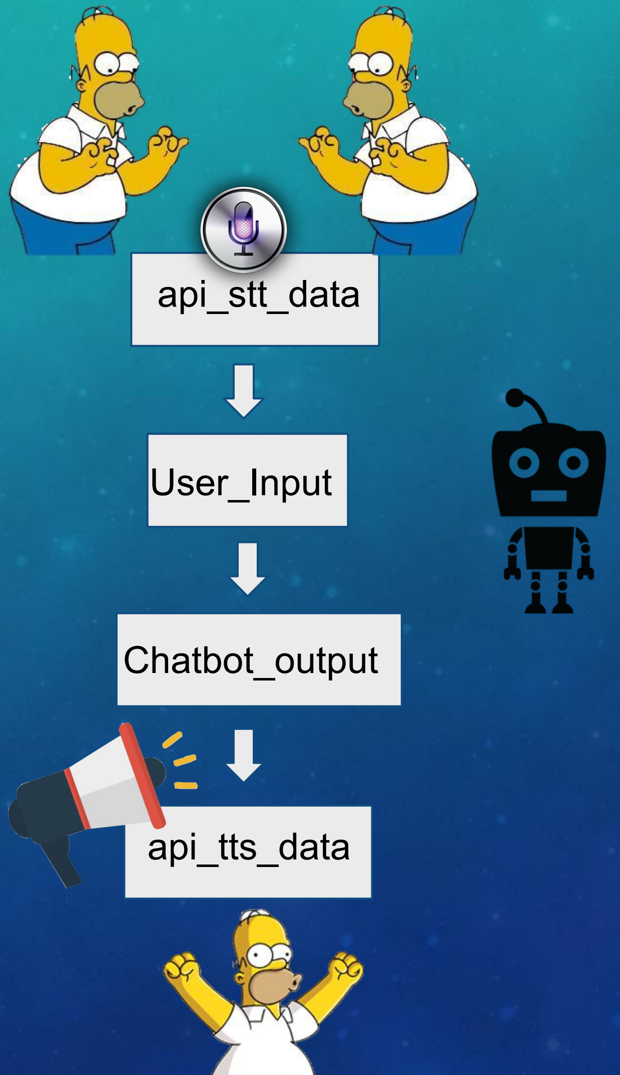
TABEL OF CONTENTS

- Speech/multimodal API
- Reinforcement learning based dialogue policy
- NN-based NLG
- Performance for simulated dialogues

Speech/multimodal API-Web Speech Api

- ## SpeechRecognition
  Speech recognition is accessed via the SpeechRecognition interface, which provides the ability to recognize voice context from an audio input and respond appropriately. Generally you'll use the interface's constructor to create a new SpeechRecognition object, which has a number of event handlers available for detecting when speech is input through the device's microphone. The SpeechGrammar interface represents a container for a particular set of grammar that your app should recognise

- ## SpeechSynthesis
  Speech synthesis is accessed via the SpeechSynthesis interface, a text-to-speech component that allows programs to read out their text content .Different voice types are represented by SpeechSynthesisVoice objects, and different parts of text that you want to be spoken are represented by SpeechSynthesisUtteranceobjects. You can get these spoken by passing them to the SpeechSynthesis.speak() method



Reinforcement learning based dialogue policy

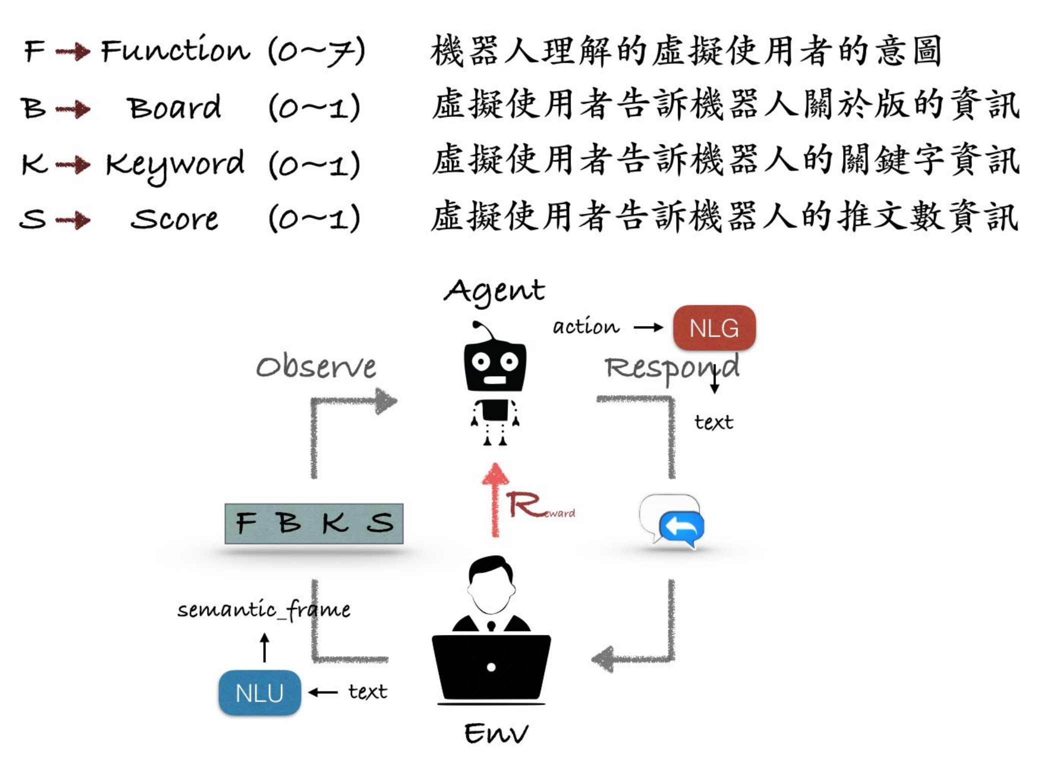
http://arxiv.org/pdf/1312.5602.pdf

參考Keras-rl & GYM 的架構

Reinforcement Learning
States:
simuated user說話經過nlu變成observation, 然而我們把一些我們認為重要的資訊(user's intent, 版, 關鍵字, 推文數)當作states 丟入reinforcement learning 的 input 讓agent 去學習.

reward:
simulated user 對比 agent 理解到的states 與 user's states 做比較來給出分數

action:
DQNagent 的Policy 是 BoltzmannQPolicy
action = self.forward(observation)
再用reward 去教導這個行為好不好



Reinforcement learning based dialogue policy

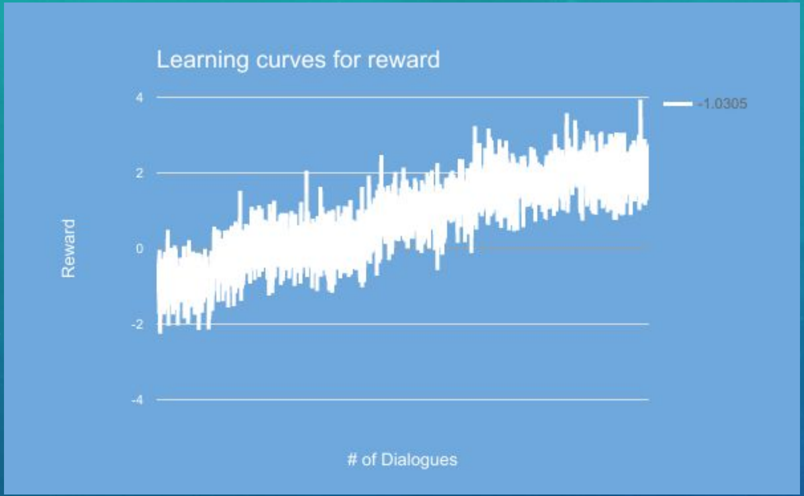
Network 架構：
第一層input layer (1, #states) #states=4
第二層 16個neurons, activation='relu'
第三層 16個neurons, activation='relu'
第四層 16個neurons, activation='relu'
第五層 #action個neurons #action=7個

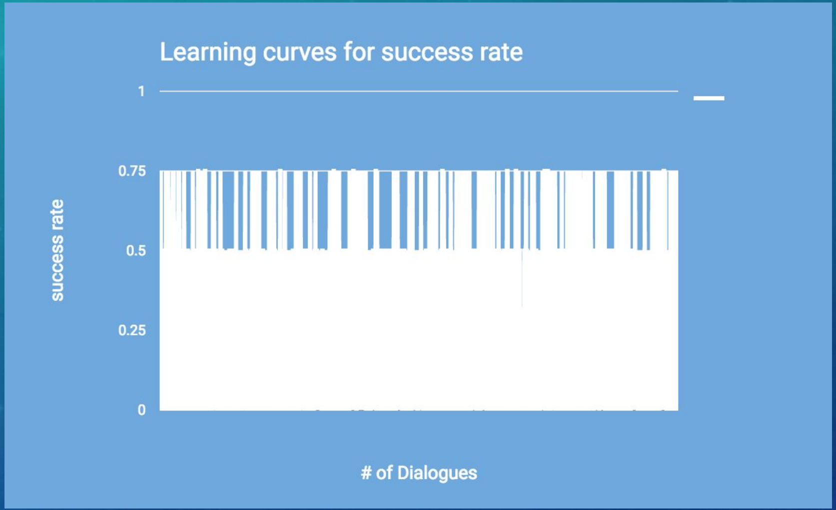
Actions:
0 request_function 詢問user的intent
1 request_board 詢問要什麼版
2 request_title 詢問關鍵字
3 request_score 詢問推文數需要多少
4 inform_board 詢問版的資訊
5 inform_post 告訴user 他想要的文章
6 read_post 唸文章
7 inform_comment 告訴user 下面的推文
8 unknown 不懂這個state 在幹麻

```
# Next, we build a very simple model.
model = Sequential()
model.add(Flatten(input_shape=(1,) + env.observation_space.shape))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(16))
model.add(Activation('relu'))
model.add(Dense(nb_actions))
model.add(Activation('linear'))
print(model.summary())
```

Reinforcement learning based dialogue policy

learning curves for reward and success rate





Reinforcement learning based dialogue policy

Reward and Sucess-rate Setting

- First, The user will check if the intent in the semantic frame is matched. If it's mathed, the user will give BOT a postive reward, otherwise, negative.
- Also, The Bot will pass the action to the user, and the user will determine if this action is appropriate in this dialogue case to give the responding reward.
  For example :
  user:可以幫我找文章嗎
  (O) BOT:請你你要找的版是？ Case 1 : the user will give a postive reward
  (O) BOT:推文數大於多少你才想看？ Case 3: Also a postive reward
  (X) BOT:你在公三小？ Case 8: A negative reward
- If the BOT has already received the information which has been told before, but it asked repeatedly, it will also get a negative reward
- If the BOT outputs the result once it get enough information, it will get a bigger positive reward
- We set the frame-level sematics as our success-rate . Currently, our BOT only supported four slots to be filled, so basically we couldn't see a clear trend in the success-rate curve.

NN-based NLG

RNN Unit:

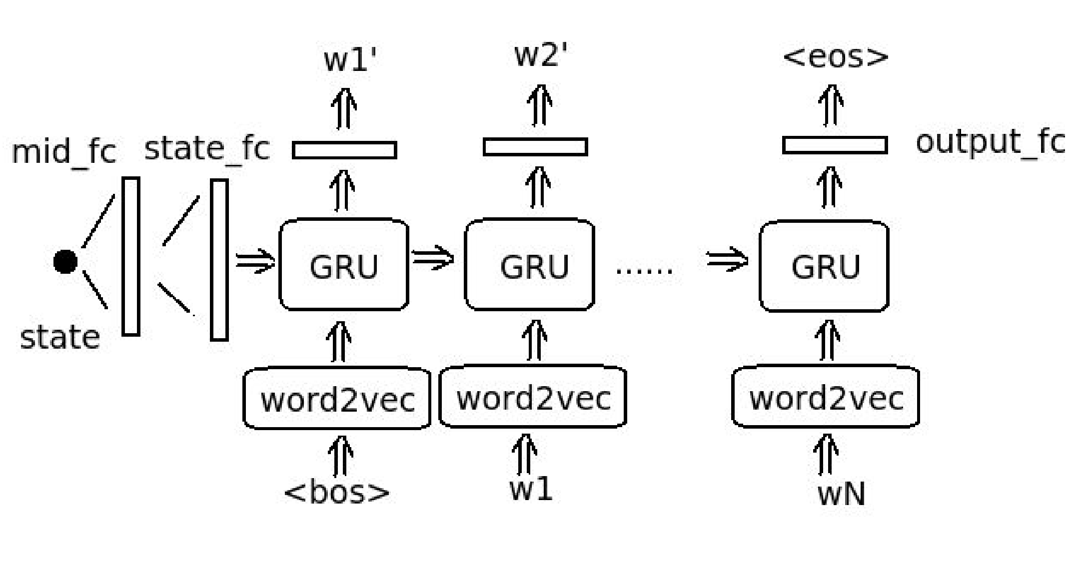
When the training data is not large, the model should not be too complex. Hence, we used GRU as our RNN unit rather than LSTM unit.

Network Structure:

1. RNN with one-layer GRU
2. State Representation: action number
3. State Input Layer: 2 fc layer
4. RNN Input: word2vec

Network Settings:

1. mid_fc: 50 nuerons
2. state_fc: 256 neurons
3. word2vec output: 256 neurons
4. GRU hidden state dimension: 256
5. output dimension: dicitionary size(116)



NN-based NLG (Training Settings)

State Noise:

Since one state can have multiple sentences, we map each sentence to a noised state rather than the original state.

The noise is a zero-mean Gaussian noise with std = 0.1.

Training Loss: word-by-word softmax with cross-entropy

Optimizer: Adam (learning rate=0.01, beta1 = 0.9, beta2=0.999)

Training Input: the input at time t is directly the one hot-code of the groundtruth at time t-1

Training Epochs: 2000

Batch_size: 20

RNN max numsteps: 30 (we mask the loss of redundant outputs)

Training Data:

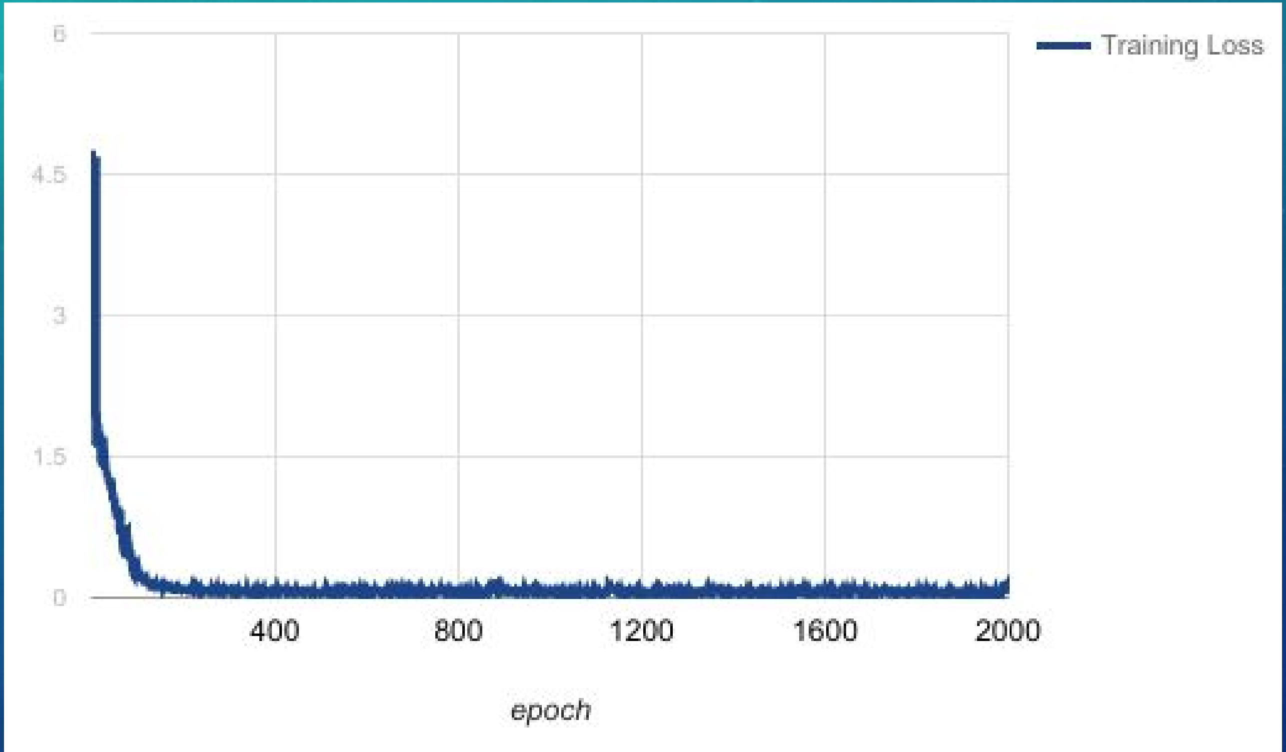
template sentences for each state, each state has approximately 4~7 sentences

(the training data can be found in “opt/nlg/template.txt”)

NN-based NLG

Final Loss: 0.0824

Training Loss:



NN-based NLG (Testing)

Diversity:

To increase the diversity of our network, we add a noise into the state input.

The noise is a zero-mean Gaussian noise with std = 0.05.

Evaluation:

For each of 9 state, we generated 500 sentences from NLG and evaluate them with the testing data.

Evaluation Toolkit: nltk.transfer.bleu_score.sentence_bleu()

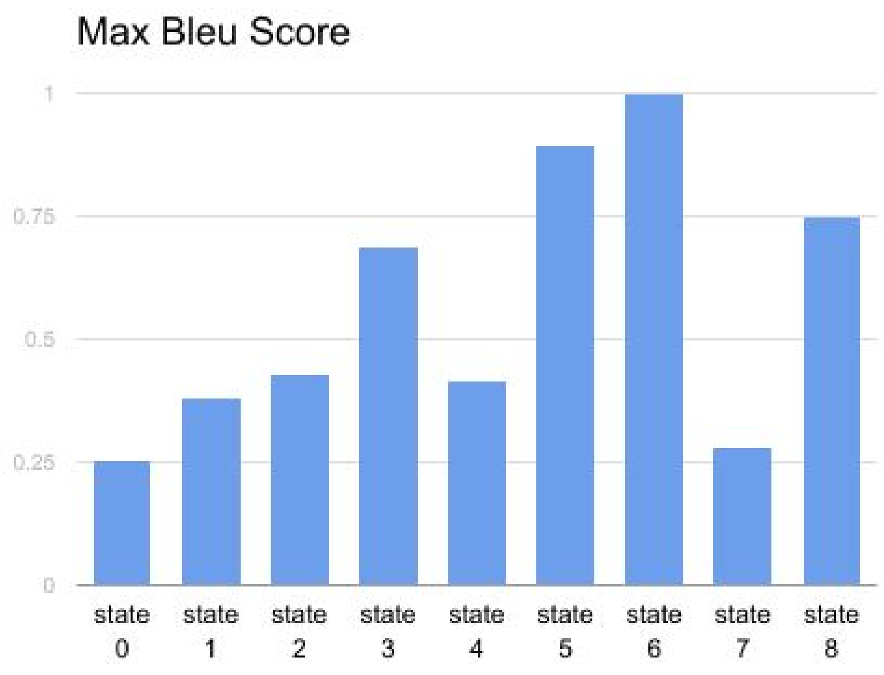
metrics: average bleu score, max bleu score

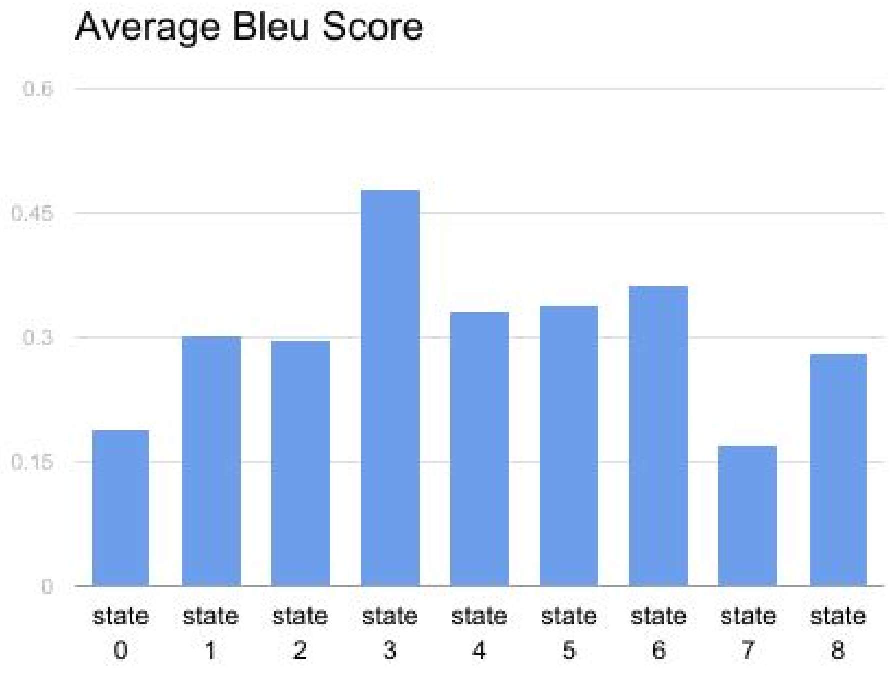
Testing Data:

template sentences for 9 states generated by another crew member who did not know the training data.

(the testing data can be found in “opt/nlg/template_test.txt”)

NN-based NLG


Max Bleu Score
1
0.75
0.5
0.25
0
state 0
state 1
state 2
state 3
state 4
state 5
state 6
state 7
state 8


Average Bleu Score
0.6
0.45
0.3
0.15
0
state 0
state 1
state 2
state 3
state 4
state 5
state 6
state 7
state 8

Performance for simulated dialogues

Show some dialogues between the simulated user and the RL agent

```
[Simulated User Say :]:我要找文章
5/5 [==============================] - 0s
5/5 [==============================] - 0s
request_post
我<unk>
要<unk>
找<unk>
文<unk>
章<unk>
[count] ->>1
self.states:{'function': 'request_post'}
```

```
[ChatBot Say :]:要大於多少推文數呢
[action] ->>3  request_score
[Simulated User Say :]:讚至少要20
5/5 [==============================] - 0s
5/5 [==============================] - 0s
request_post
讚<unk>
至<unk>
少title
要<unk>
20score
```

the reward is 1.1, which means the response is reasonable

```
[ChatBot Say :]:你希望查詢哪個版呢
[action] ->>1  request_board
[success rate]1.0
[dialogue_reward]3.695
[action] ->>5  inform_post
[Simulated User Say :]:我需要的是joke
6/6 [==============================] - 0s
6/6 [==============================] - 0s
request_post
我<unk>
需<unk>
要<unk>
的<unk>
是<unk>
jokeboard
```

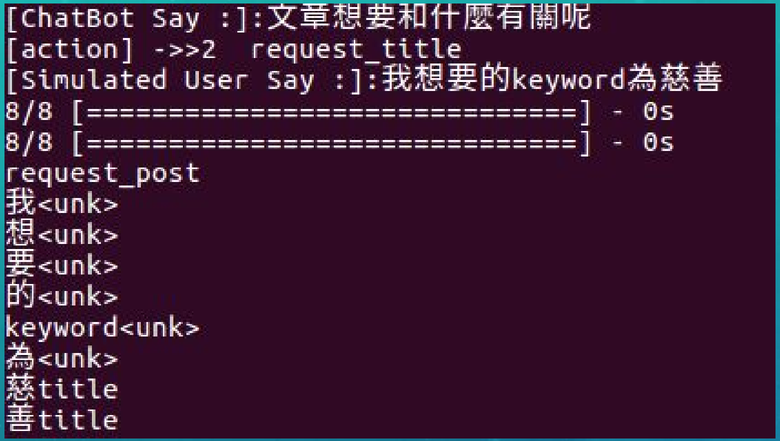
the reward is 1.1, which means the response is reasonable

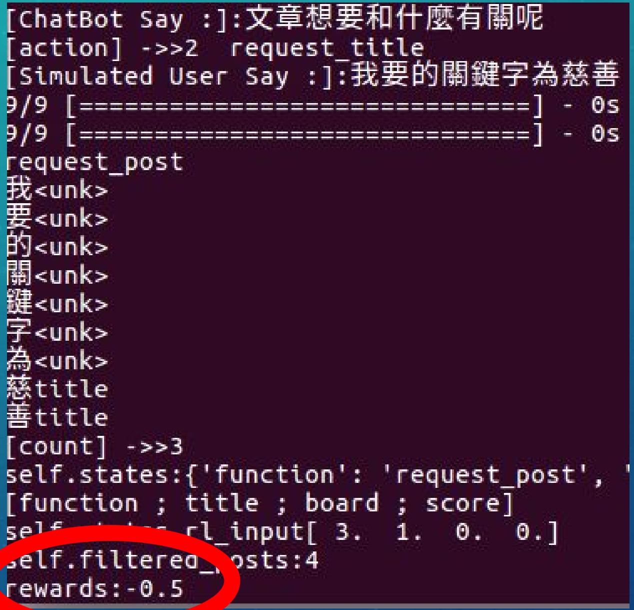
```
[ChatBot Say :]:文章想要有關什麼呢
[action] ->>2  request_title
[Simulated User Say :]:我想要keyword為巫師
7/7 [==============================] - 0s
7/7 [==============================] - 0s
request_post
我<unk>
想<unk>
要<unk>
keyword<unk>
為<unk>
巫title
師title
```

the reward is 1.1, which means the response is reasonable

Performance for simulated dialogues

Show some dialogues between the simulated user and the RL agent





The example of the punishment reward when the BOT do the same action during training.