*I*ntelligent *C*onversational *B*ot 2017

PTTBOT2: MINING PTT SMARTLY

C.H, Yu H.W, Hsu C.K, Hsieh P.C, Zheng

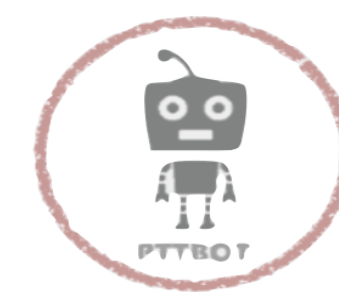**國立臺灣大學 National Taiwan University**

Demo link: http://140.112.251.159:5555   QR Code:

Overview

If you are not so familiar with PTT, don't worry! Our PTT-BOT is here for you.
With it, you are going to be a professional villagers!(鄉民)
Here are some functionalities you must try!
(1)「BOT Full Of Emotion 」
 -The greeting and the reply of our BOT will adjust based on the facial expression of the user.
(2)「Search Desired Posts For U 」
 - Our BOT can search for related posts according to the information provided by users.
(3)「Search Desired Board For U 」
 -Often having no idea what board it has/which board to find the things/topics you want ? Our BOT can recommend related board according to the keywords.
(4)「Track Users U R Interested In」
 -Our BOT can find shared accounts of the same user. (找分身)
(5)「Customized Features and So on…」
 -Wait for you to Discover !
(6)「All Chat-BOTs IN THIS ONE PTT-BOT 」
 -Support multi-domain Searching :Moive, Music, Course, Stock, Food, Job...etc

[User]幫我找分身
[BOT]請問你要找的作者是？
[User]Obov
[BOT]經過縝密的思考後....

** Supported APIs: Speech, Vision, and Emotion

Speech Signal

Text Input
請給我有關統神的文章

Speech Recognition

Language Understanding (LU)
• Domain Identification
• User Intent Detection
• Slot Filling

Semantic Frame
request_post
key_word='統神'

Natural Language Generation (NLG)

Dialogue Management (DM)
• Dialogue State Tracking
• Dialogue Policy

Text response
請問你想在哪個版查詢呢?

System Action/Policy
request_board

Backend Database

Ontology

We crawled and collected data from the top five boards(**Gossiping, Sex, Lol, Joke, NBA**) on PTT(https://www.ptt.cc/bbs/hotboards.html) , and built our database including the fields below.
Each entity indicates a post on PTT.

• **Backend Tables :**

board	title	author	content	comments {state,message,id,date}	push {all,score,good,bad,none}	date	ip	url
Gossiping …….	[新聞] …	Clean	花七億公帑…	….	….	….	….	….
Sex …….	[閒聊]	Pttsex	喜歡當姊姊….	….	….	….	….	….

- In the dictionary of comments:[state:推/噓/→ ,message:訊息, id: 推文者, date:推文時間]
- In the dictionary of push:[all: 此篇推文總數, score:推-噓, good:推總數, bad:噓總數, none:→ 總數]

• **Overview of DB Table**
 -The size of DB: In total, We have nearly **50K** posts from this five boards.
 -Number of columns: As you can see in the table, our DB has **16** columns.
 -Number of slots: We support **11** slots in the semantic frame, expect comments and date.
 -Number of intents: PTTBOT can support up to **10** intents, to make most of the slots could be be searched possibly.

• **What's more ?**
-In fact, Our BOT even has the functionality of **real-time searching** on PTT (but responding time is slower), which offers you the latest information and makes you a proficient in PTT.
(專業的五樓,掌握最新趨勢的首選Chat-Bot)

Dialogue Management

• **Model architecture: Reinforcement Learning (RL)-Deep-Q learning (DQN)**
 -We take Keras-rl and GYM as our reference to build our model.
 -RL states: The NLU turns the sentences from simulated user (environment) into observation , in which we extract our desired states as the input of the agent.
 -Action: The policy in DQN agent is Boltzmann Q policy, and the behavior is trained by the reward given by the simulated user.

• **User Simulator**
 -When an instance of simulated user is created, it will own a semantic frame itself, where the intent is randomly picked, and in each intent, the instance will have a corresponding semantic frame.
 - For example, When the simulated user wants to request post, the semantic frame of this instance includes (i) board (ii) key words (iii) # of pushes.

Diagram of DQN

Agent
Observe
action → NLG
Respond
text
F B K S
R
semantic_frame
NLU ← text
Env

• **Success rate / Reward**
 -Setting:
(a) The user will check if the intents in the semantic frame is matched. If it's matched, the user will give BOT a positive reward, otherwise, negative.
(b) 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.
(c) If the BOT has already received the information which has been told before, but it asked repeatedly, it will also get a negative reward.
(d) If the BOT outputs the result once it gets enough information, it will get a relatively positive reward.

-Trend of the learning curves for rule-based and RL agents (Rewards) with 20K dialogues

Learning curves for reward

＊Difference Between Rule-based and RL-based model

RL
[User]:我要找文章
[BOT]:要大於多少推文呢?
[User]:讚至少要20
[BOT]:你希望查詢哪個版?
[User]:我需要的是joke
[BOT]:文章想要關於什麼
[User]:我要關於卡提諾
[BOT]:文章想要關於什麼

Rule-based
[User]:我要找文章
[BOT]:你希望查詢哪個版?
[User]:我需要的是joke
[BOT]:文章想要關於什麼
[User]:我要關於卡提諾
[BOT]:要大於多少推文呢?
[User]:讚至少要20
[BOT]: ….Result….

Language Understanding

• **Model Architecture**
 -We have two separate model to determine the intent and do the slot-filling respectively.
 (a) **Intent Prediction**: Feed-Forward recurrent neural networks
  -Structure: Feed-Forward RNN with hidden size = 100.
 (b)**Slot Filling :**Bidirectional recurrent neural networks (**BRNN**)
  -Structure: Bidirectional RNN with hidden size = 100
  -In this structure, the output layer can get information from past and future states, which is especially useful for slot-filling.

Diagram of (b)   Diagram of (a)

• **Data Collection**
 -We created more than 10 templates for each intents, and collected the key words from PTT to make the training data diverse.
 -Training size: In total, We have about **1.6M** sentences for training.
 -Testing size: 1) from training data: **8K** 2) from real user: **50**

• **Performance On Testing data (Accuracy is defined as Frame Matched Rate)**
 -If the testing data is extracted from training data (created by the template), the accuracy could be up to **100%.**
 -If the testing data is from real users, the accuracy is about **84%**

Natural Language Generation

• **Model architecture : NN-based**
 -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: (a) RNN with one-layer GRU (b) State Representation: action number (c) State Input Layer: 2 fc layer (d) RNN Input: word2vec
 * More details about our model: **http://ppt.cc/TW1NU**
• **Data collection**
 -State noise: Since one state can have multiple sentences, we map each sentence to a noised state rather than the original state.
 -:Training size Template sentences for each state, each state has approximately 4-7 sentences.
 -Training setting: (a) Training Loss: word-by-word Softmax with cross-entropy (b) Optimizer: Adam (c) Training Input: the input at time t is directly the one hot-code of the ground truth at time t-1 (d) Training Epochs: 2000 (e) Batch size: 20 (f) RNN max num_steps: 30.
 -Testing data: Template sentences for each state generated by other user who didn't know the training data. (**real human**)
 -**Evaluation: Average bleu score and max bleu score**

• **Performance**

Max Bleu Score

Average Bleu Score