Beihang University
School of General Engineering

Beijing, May 11th, 2021
Lin Wei and Yida Ding

Data Management and Artificial Intelligence Lab Class 10

Task 1 ***Necessary elements preparation*** *(10 minutes)*

(1) Download and unzip the package of *ngrok* for your OS (windows or linux).

(2) Install three python packages named *waitress*, *falcon* and *xmltodict* with pip (Just type *"pip install xxx"*).

Task 2 ***Test a silly chatbots for wechat*** *(25 minutes)*

In the following steps, you will build and test a silly bot for wechat. Are you ready? Go!

(1) Download the code package from m3nets and run *server.py* with spyder. Then run the command *"./ngrok http 8000"* (for linux) or *"ngrok http 8000"* (for windows) in the terminal (Donot forget to set the path correctly!). Then, you will see something like Figure 13.

Figure 13

Figure 14

Figure 15

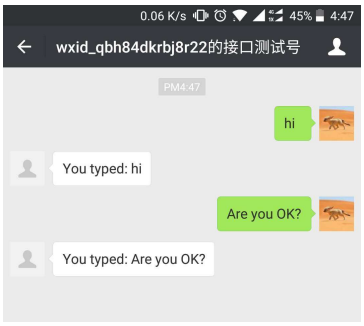

Figure 16

(2) Register a wechat sandbox which provides a free service to access wechat API. The link is *https://mp.weixin.qq.com/debug/cgi-bin/sandbox?t=sandbox/login*, just scan the QR code to log in and turn on the option for voice recognition. The desirable picture is shown in Figure 14.

(3) Type the link *"http://xxxx.ngrok.io/wechat"* after *URL*. Here the link before *"/wechat"* is the link in the line *"forwarding"* in your terminal (see Figure 13). Type *"wechat_token"* after *Token*. You can see Figure 15 as a result.

(4) Click the submit button(if not success, try again). Use your wechat to follow this test account. Then you can try the service by typing in the chat window. The result should be like Figure 16.

Task 3 ***Advanced web services in wechat*** *(25 minutes)*

In this task, we would like to provide users with some advanced function like automatic chatting(not echo), or recognize the voice as words.

(1) *Eliza* is a simple and good program for automatic chatting. It outputs a suitable response for an input sentence, just like Figure 17. Please use the *elizax.py* to respond automatically to users' words.

Tip: Focus on the function *on_post()* in the file *"silly_echo_bot_python.py"*. After you finish the code, please restart *"server.py"* and *"ngrok"*. The results should be like Figure 18.

```
In [7]: import eliza

In [8]: aa=eliza.eliza()

In [9]: aa.respond('How are you?')
Out[9]: "What is it you're really asking?"

In [10]: aa.respond('I am fine. Thank you.')
Out[10]: 'How long have you been fine. thank you.?'
```

Figure 17

(2) To set up a calculator that can calculate the string(e.g. $(1+2)*3=9$), we use the function *eval()* in python to obtain this. To do so, we need to define a function which indicates whether the input string is number or mathematical functor ("1,2,3...,+,-,*,/,(,)"). The results should be like Figure 19.

Figure 18

Figure 19

Task 4 ***Agents: Missionaries and Cannibals*** *(35 minutes) Model and solve the Missionaries and Cannibals problem from the lecture (Slide 40) using Python.*

1. Define a class *State* which models a single state
2. Define a member function *getActions()* which returns the set of available (=legal) actions for a given state S

3. Define a member function *isGoal()* which decides whether a state is a goal state of the problem.
4. Declare an initial state S_{init}
5. Define a state-transition function, which computes the successor state of a given initial state and a legal action
6. Implement BFS over the state space and print the identified solution (=sequence of actions)
7. Implement DFS over the state space and print the identified solution (=sequence of actions)