Chatbot
Lab 7

[Getting started]

Today we are going to learn ChatterBot, a machine-learning based conversational dialog engine. For a full description of ChatterBot, please visit
https://github.com/gunthercox/chatterbot

In this lab, we will train a simple chatbot with an officially provided corpus. Given input text, ChatterBot can generate responses based on collections of known conversations. As interacting with users, ChatterBot saves the input text and responses.

First, install ChatterBot with

```
pip install chatterbot
```

To use the high-quality official corpora, install the corpus library with

```
pip install chatterbot_corpus
```

Run “import chatterbot” and “import chatterbot.corpus” to validate installation.

[Functions]

Let us examine the functions one by one.

First, we try building a chatbot.

Basic Usage

```
#encoding=utf-8
from chatterbot import ChatBot
from chatterbot.trainers import ChatterBotCorpusTrainer

bot = ChatBot('test') # You can rename it whatever you want
trainer = ChatterBotCorpusTrainer(bot)
trainer.train('chatterbot.corpus.chinese') # Load conversations

while True:
    text = input('You: )
```

```
    print('Bot: %s' % bot.get_response(text))
```

You can try different options according to the ChatterBot's document. You can change its storage adapter, confidence threshold for responses and the default response, etc., or even try some fancy usages.

Adding more conversations

When using the provided corpus, you would probably get conversations like

```
You: 你好
Bot: 你好吗?
You: 还可以
Bot: 股市
You: 股市怎么了?
Bot: 与人类一样,除了我们缺乏所有的情感,梦想,愿望,创造力,野心,尤其是主观性。
```

The official training corpus is small, merely covering greetings and some specific conversations, as you can check at https://github.com/gunthercox/chatterbot-corpus/tree/master/chatterbot_corpus/data/chinese. To make conversations more meaningful and interesting, we can train the bot with more data.

```
from chatterbot.trainers import ListTrainer
list_trainer = ListTrainer(bot)
list_trainer.train(['还可以', '那太好了'])
```

We can also incorporate conversations from a larger corpus. Same samples from the *xiaohuangji* corpus are provided here. Note that the corpus is not very clean, as its name suggests…

```
lines = open('/data/cs310/baibing/lab7/xiaohuangji.txt').readlines()
list_trainer.train(lines)
```

Dealing with time and math

We can let the chatbot do some simple calculation and tell us the time.

```
bot = ChatBot(
    logic_adapters=[
        'chatterbot.logic.MathematicalEvaluation',
        'chatterbot.logic.TimeLogicAdapter',
```

```
    ]
)
>>>print(bot.get_response('what time is 1 * 2 + 100?'))
1 * 2 + 100 = 102
>>>print(bot.get_response('what time is it?'))
The current time is 09:04 PM
```