COSC310
Interactive Conversational Agent
Test Cases

Group 23

The file **test_chatbot.py** is used to run unit testing on the bot. The file can be run directly after the Stanford NLP server is started. There is a set up process that will load in the **intent.json** file to grab some expected responses that will be tested. There are a total of seven tests that are run, covering major parts of the bots functionality.

```
def setUp(self):
    self.l = load("intents.json")
    self.data = self.l.getData()
    for t in self.data["intents"]:
        if t['tag'] == "greeting":
            self.responses = t['responses']
        if t["tag"] =="music":
            self.responsesmusic = t['responses']
```

Test Basic Chat Functionality

The test_chat function tests basic chat inputs and responses. It will assert true if an input matches with an expected response. For this function we've decided to check responses for greetings. As the bot is programmed to throw out a random relative response from the matching tag, we must check if the response we receive is inside the list of responses. If it is, the function will assert it as true.

```
def test_chat(self):
    self.assertTrue(chatbot.chat("How are you") in self.responses)
    self.assertTrue(chatbot.chat("Is anyone there?") in self.responses)
    self.assertTrue(chatbot.chat("hello") in self.responses)
    self.assertTrue(chatbot.chat("good day") in self.responses)
    self.assertTrue(chatbot.chat("whats up?") in self.responses)
```

Testing Tags

Each response that we receive is related to a tag. This tag is used to output the response, so we can check if the input that we give matches the expected tag. Using assert equals we check if chatbot.tag is equal to the relevant tag.

```
def test_tag(self):
    chatbot.chat("How are you")
    self.assertEqual(chatbot.tag,"greeting")
    chatbot.chat("see you later")
    self.assertEqual(chatbot.tag,"goodbye")
    chatbot.chat("how old")
    self.assertEqual(chatbot.tag,"age")
```

Testing Sentiment Analysis

Sentiment analysis works by returning a statement relevant to an input. This is tested by using assert equals as the responses are all hard coded, and work based on a scale from 0-3.

```
def test_sentiment(self):
    self.assertEqual(chatbot.chat("I love your work"),"Glad to hear you really like that.")
    self.assertEqual(chatbot.chat("I hate you"),"I can understand that.")
    self.assertEqual(chatbot.chat("I like this conversation"),"I fully agree.")
```

Testing Capitalization

This is a simple one, as all that's needed is to input the same statement with different capitalization. We then check if the response we receive is in the expected responses list.

```
def test_capitalization(self):
    self.assertTrue(chatbot.chat("hOW aRE yOU") in self.responses)
    self.assertTrue(chatbot.chat("HOW ARE YOU") in self.responses)
    self.assertTrue(chatbot.chat("how are you") in self.responses)
```

Testing Punctuation

Another simple test. Similar to the one above, we input the same statement, with different punctuation and check if the expected response is in the responses list.

```
def test_punctuation(self):
    self.assertTrue(chatbot.chat("how are you.") in self.responses)
    self.assertTrue(chatbot.chat("how are you!") in self.responses)
    self.assertTrue(chatbot.chat("how are you?") in self.responses)
```

Testing Synonyms

For this test, we must put in similar sentences, with only a few words swapped around that are not in our expected inputs list. Then we can check if the response we receive is still in the responses list, telling us that replacing with synonyms worked.

```
def test_synonyms(self):
    self.assertTrue(chatbot.chat("what type of music do you listen to") in self.responsesmusic)
    self.assertTrue(chatbot.chat("what tunes do you listen to") in self.responsesmusic)
```

Testing Out of Scope Responses

This test will check that the bot gives an appropriate response if it can't understand what the input is. It asserts true if the expected response is in chatbot.others list.

```
def test_failResponses(self):
    self.assertTrue(chatbot.chat("Who is Gahlran?") in chatbot.others)
    self.assertTrue(chatbot.chat("do you play among us?") in chatbot.others)
```