Assignment 1

Artificial Intelligence

CISC 352, Winter, 2017

Due Friday, February 3, 2017 before Midnight

Preamble

The Turing Test resulted in a number of computer programs designed to mimic human conversation. The first, and probably the best known, system to simulate conversation was ELIZA, developed by Joseph Weizenbaum in the mid-1960s. The program, named after the ingénue in George Bernard Shaw's Pygmalion, featured various scripts. The most well-known of ELIZA's scripts, the DOCTOR script, was designed to emulate an initial appointment with a Rogerian psychotherapist. As such, the program had only to reflect back the patient's statements. For instance, a typical conversation may be:

I am the psychotherapist. Please, describe your problems.

Men are all alike.

In what way?

Well, my boyfriend made me come here.

Your boyfriend made you come here?

He says I'm depressed much of the time.

I am sorry to hear you are depressed.

It's true. I am unhappy.

Do you think coming here will help you not to be happy?

I need some help, that much seems certain.

What would it mean to you if you got some help?

Perhaps I could learn to get along with my mother.

Tell me more about your family.

ELIZA is fundamentally a pattern matching program. For instance, first-person pronouns (I, My, Me) are mapped to second-person pronouns (You, Your, You) and vice-versa. It is used to reflect a statement back against the user.

Assignment

1. Read Weizenbaum's paper detailing ELIZA. It is located on the #assignment1 channel on Slack. It's also on the ACM digital library (http://dl.acm.org/citation.cfm?id=365168&dl=ACM&coll=DL&CFID=879382445&CFTOKEN=79194444). I apologize that it's blurry…it was written in the 1960s. ☺
2. Develop a chatbot in the form of an ELIZA/DOCTOR script. This can be in any programming language of your choosing, so long as I can run the program and enter statements to chat with the bot.
3. Turn in the chatbot, along with a technical document and a README describing how to run the program. You will be graded on the quality of the technical document, but not on the quality of the README (unless I can't get your program to run using your instructions).
4. The technical document should provide details about the algorithm you implemented, along with each function and their purpose. The document is required because I allow any high-level language. It's also good practice to be able to explain your algorithm. The key things to this document are that you explain your algorithm well and present it in the best manner possible.
    a. For instance:

> First, the program prints the initial prompt to the user, "Hello. How are you feeling today?". Then, we enter a loop asking the user for input and passing what the user enters to the "analyze" function to get the therapist's response. If, at any point, the user types "quit", we break out of the loop, and the program exits.
>
> The program contains two global arrays, "reflections" and "psychobabble". The "reflections" array maps first-person pronouns (I, Me, My) to second-person pronouns and vice-versa. It is used to "reflect" a statement back to the user. The "psychobabble" array is a list of lists, where the first element is a regular expression that matches the user's statements, and the second element is a list of potential responses. Many of the potential responses contain placeholders that can be filled in with fragments to echo the user's statements.
>
> In the "analyze" function, we iterate through the regular expressions in the "psychobabble" array, trying to match each one with the user's statement, from which we have stripped the final punctuation. If we find a match, we choose a response template randomly from the list of possible responses associated with the matching pattern. Then, we interpolate the match groups from the regular expression into the response string, calling the "reflect" function on each match group first.
>
> In the "reflect" function, we make the statement lowercase, then we tokenize it by splitting on whitespace characters. We iterate through the list of tokens and, if the token exists in our "reflections" dictionary, we replace it with the value from the dictionary. So, "I" becomes "you", "your" becomes "my", and so on.

b. Note that unlike the sample text above, it would be a good idea to place your code between paragraphs. You can append the code to the end of the technical document, but it is much easier to understand the algorithm if you explain a function, and then show that function. Please, compose the document to the best of your ability.
c. Note that if you turn in exactly this text, it will not be looked upon favorably. While your algorithm may be similar, don't copy and paste the text. ☺
d. Also, note that a sample python script is provided in this document. Weizenbaum implemented a much more complicated script. If you go above and beyond to match Weizenbaum's implementation, it will be looked upon highly.
e. And a last note, to keep the chatbots consistent, please use the sample prompt for input: "Hello. How are you feeling today?"

<u>**Grading**</u>

There are four parts to the grading. The first three parts make up the primary grade for the assignment. Part 4 is for bonus points added to the total.

1. Part 1:
    a. The first part is a competition. You will be randomly placed into groups of 5, as evenly as possible. So, if I receive 85 assignments (approximately the number of students in the class as of this writing), you will be randomly placed in one of 17 groups. I realize that it's possible I could receive 86 assignments or so on. If that is the case, I will grade in a manner that maximizes fair competition (I have developed an algorithm for this purpose).
    b. The chatbots will then compete. All the chatbots will receive the same text passage.
    c. The best chatbot in the group receives 5 points, the next receives 4 points, the next 3, the next 2, and the worst chatbot in the group receives 1 point. So, for this round a total of **5 points** are possible.
    d. The groups of 5 are then rearranged. The winners of the previous round are placed against each other, second place against the other second place, third against third, and so on. The scores are tallied in the same manner as the previous round, so a total of **5 points** are possible this round. Each round has a new passage of text.
    e. This is done for five rounds total, giving a maximum of **25 points** for Part 1.
2. Part 2:
    a. The second part is a standalone performance evaluation. I will generate five short passages of text that will attempt to behave as someone going to a Rogerian psychotherapist for an initial appointment. I will run your chatbot against each of the five passages. Then, I will calculate a score based on the following equation:

    $$\frac{\#sensibleResponses}{\#totalResponses} \times 25$$

    So, the max you can get is **25 points** for Part 2.

An example of a non-sensible response would be the following (in bold):

How long have you been writing?
Always, but I've been writing seriously since I was eighteen.
You've been writing since you were eighteen?
Yeah.
Can you elaborate on that?
I started writing in university.
You started writing in university?
Yeah.
Why do you say that "you started writing in university"?

Note that ELIZA is easy to fool if you don't play along. My passages are intended to play along and not try to trick the chatbot. Also, if ELIZA gets stuck a couple of times in a row, I will reinitiate the conversation by going off-script to get it back on track so that you are not counted off severely.

3. Part 3:
   a. The third part is a grade on your technical document and algorithm. This is worth **50 points**. I've weighted this heavily so that if you didn't do so hot in the competition, so long as you did well on explaining your algorithm and the overall output of your code, you still get a good grade. The grade will include grammar, spelling, presentation quality, and the description of your algorithm. I should be able to know exactly how your algorithm works, and it should definitely not be copy-pasted from this document.
4. Part 4:
   a. Bonus points. If your chatbot stands out in some way, you earn points. If your algorithm is exceptionally well thought out, you earn points. If the conversation is very realistic, you'll earn bonus points. The maximum bonus points you can earn is **15**, but you cannot go over 100 points. So, if you have 100 already, you aren't earning 115.

The total points you can earn for this assignment is **100**.

This assignment is 17% of your total grade.

Sample Script

The following is a sample ELIZA script written in Python. It uses the algorithm discussed earlier in the provided sample technical document.

```
import re
import random

reflections = {
   "am": "are",
   "was": "were",
   "i": "you",
   "i'd": "you would",
   "i've": "you have",
   "i'll": "you will",
   "my": "your",
   "are": "am",
   "you've": "I have",
   "you'll": "I will",
   "your": "my",
   "yours": "mine",
   "you": "me",
   "me": "you"
}


psychobabble = [
   [r'I need (.*)',
    ["Why do you need {0}?",
     "Would it really help you to get {0}?",
     "Are you sure you need {0}?"]],

   [r'Why don\'?t you ([^\?]*)\??',
    ["Do you really think I don't {0}?",
     "Perhaps eventually I will {0}.",
     "Do you really want me to {0}?"]],

   [r'Why can\'?t I ([^\?]*)\??',
    ["Do you think you should be able to {0}?",
     "If you could {0}, what would you do?",
     "I don't know -- why can't you {0}?",
     "Have you really tried?"]],

   [r'I can\'?t (.*)',
    ["How do you know you can't {0}?",
     "Perhaps you could {0} if you tried.",
     "What would it take for you to {0}?"]],

   [r'I am (.*)',
    ["Did you come to me because you are {0}?",
     "How long have you been {0}?",
     "How do you feel about being {0}?"]],
   [r'I\'?m (.*)',
    ["How does being {0} make you feel?",
     "Do you enjoy being {0}?",
     "Why do you tell me you're {0}?",
     "Why do you think you're {0}?"]],

   [r'Are you ([^\?]*)\??',
    ["Why does it matter whether I am {0}?",
     "Would you prefer it if I were not {0}?",
     "Perhaps you believe I am {0}.",
     "I may be {0} -- what do you think?"]],

   [r'What (.*)',
    ["Why do you ask?",
     "How would an answer to that help you?",
     "What do you think?"]],

   [r'How (.*)',
    ["How do you suppose?",
     "Perhaps you can answer your own question.",
     "What is it you're really asking?"]],

   [r'Because (.*)',
    ["Is that the real reason?",
     "What other reasons come to mind?",
     "Does that reason apply to anything else?",
     "If {0}, what else must be true?"]],

   [r'(.*) sorry (.*)',
    ["There are many times when no apology is needed.",
     "What feelings do you have when you apologize?"]],


[r'Hello(.*)',
    ["Hello... I'm glad you could drop by today.",
     "Hi there... how are you today?",
     "Hello, how are you feeling today?"]],

   [r'I think (.*)',
    ["Do you doubt {0}?",
     "Do you really think so?",
     "But you're not sure {0}?"]],

   [r'(.*) friend (.*)',
    ["Tell me more about your friends.",
     "When you think of a friend, what comes to mind?",
     "Why don't you tell me about a childhood friend?"]],

   [r'Yes',
    ["You seem quite sure.",
     "OK, but can you elaborate a bit?"]],
   [r'(.*) computer(.*)',
    ["Are you really talking about me?",
     "Does it seem strange to talk to a computer?",
     "How do computers make you feel?",
     "Do you feel threatened by computers?"]],

   [r'Is it (.*)',
    ["Do you think it is {0}?",
     "Perhaps it's {0} -- what do you think?",
     "If it were {0}, what would you do?",
     "It could well be that {0}."]],

   [r'It is (.*)',
    ["You seem very certain.",
     "If I told you that it probably isn't {0}, what would you feel?"]],

   [r'Can you ([^\?]*)\??',
    ["What makes you think I can't {0}?",
     "If I could {0}, then what?",
     "Why do you ask if I can {0}?"]],

   [r'Can I ([^\?]*)\??',
    ["Perhaps you don't want to {0}.",
     "Do you want to be able to {0}?",
     "If you could {0}, would you?"]],

   [r'You are (.*)',
    ["Why do you think I am {0}?",
     "Does it please you to think that I'm {0}?",
     "Perhaps you would like me to be {0}.",
     "Perhaps you're really talking about yourself?"]],

   [r'You\'?re (.*)',
    ["Why do you say I am {0}?",
     "Why do you think I am {0}?",
     "Are we talking about you, or me?"]],

   [r'I don\'?t (.*)',
    ["Don't you really {0}?",
     "Why don't you {0}?",
     "Do you want to {0}?"]],
     [r'I feel (.*)',
    ["Good, tell me more about these feelings.",
     "Do you often feel {0}?",
     "When do you usually feel {0}?",
     "When you feel {0}, what do you do?"]],

   [r'I have (.*)',
    ["Why do you tell me that you've {0}?",
     "Have you really {0}?",
     "Now that you have {0}, what will you do next?"]],
   [r'I would (.*)',
    ["Could you explain why you would {0}?",
     "Why would you {0}?",
     "Who else knows that you would {0}?"]],

   [r'Is there (.*)',
    ["Do you think there is {0}?",
     "It's likely that there is {0}.",
     "Would you like there to be {0}?"]],

   [r'My (.*)',
    ["I see, your {0}.",
     "Why do you say that your {0}?",
     "When your {0}, how do you feel?"]],

   [r'You (.*)',
    ["We should be discussing you, not me.",
     "Why do you say that about me?",
     "Why do you care whether I {0}?"]],

   [r'Why (.*)',
    ["Why don't you tell me the reason why {0}?",
     "Why do you think {0}?"]],

   [r'I want (.*)',
    ["What would it mean to you if you got {0}?",
     "Why do you want {0}?",
     "What would you do if you got {0}?",
     "If you got {0}, then what would you do?"]],

   [r'(.*) mother(.*)',
    ["Tell me more about your mother.",
     "What was your relationship with your mother like?",
     "How do you feel about your mother?",
     "How does this relate to your feelings today?",
     "Good family relations are important."]],

   [r'(.*) father(.*)',
    ["Tell me more about your father.",
     "How did your father make you feel?",
     "How do you feel about your father?",
     "Does your relationship with your father relate to your feelings today?",
     "Do you have trouble showing affection with your family?"]],

   [r'(.*) child(.*)',
    ["Did you have close friends as a child?",
     "What is your favorite childhood memory?",
     "Do you remember any dreams or nightmares from childhood?",
     "Did the other children sometimes tease you?",
     "How do you think your childhood experiences relate to your feelings today?"]],
       [r'(.*)\?',
        ["Why do you ask that?",
         "Please consider whether you can answer your own question.",
         "Perhaps the answer lies within yourself?",
         "Why don't you tell me?"]],

       [r'quit',
        ["Thank you for talking with me.",
         "Good-bye.",
         "Thank you, that will be $150.  Have a good day!"]],

       [r'(.*)',
        ["Please tell me more.",
         "Let's change focus a bit... Tell me about your family.",
         "Can you elaborate on that?",
         "Why do you say that {0}?",
         "I see.",
         "Very interesting.",
         "{0}.",
         "I see.  And what does that tell you?",
         "How does that make you feel?",
         "How do you feel when you say that?"]]
   ]

   def reflect(fragment):
       tokens = fragment.lower().split()
       for i, token in enumerate(tokens):
           if token in reflections:
               tokens[i] = reflections[token]
       return ' '.join(tokens)


   def analyze(statement):
       for pattern, responses in psychobabble:
           match = re.match(pattern, statement.rstrip(".!"))
           if match:
               response = random.choice(responses)
               return response.format(*[reflect(g) for g in match.groups()])


   def main():
       print "Hello. How are you feeling today?"

       while True:
           statement = raw_input("> ")
           print analyze(statement)

           if statement == "quit":
               break


   if __name__ == "__main__":
       main()
```