

# Regular Expression Based Agents for Online Collection of Human-Chatbot Interactions

Lin Lue, Luis Fernando D'Haro

School of Infocomm Technology, Ngee Ann Polytechnic

llue97@hotmail.com

Institute for Infocomm and Research, Agency for Science, Technology and Research

luishe@i2r.a-star.edu.sg

**Abstract.** This paper describes various conversational agents based on regular expressions implemented in Python, providing samples of conversations with said agents; these agents are also released as examples on how to quickly implement agents that can connect with WebChat (an online platform for crowd sourcing human-chatbot conversations and their annotations).

## 1 General Description

In order to collect human-chatbot dialogs and to provide developers quick wrapper examples to connect their existing chatbots to an online platform for data collection and annotation, we implemented the following 4 chatbots which are based on regular expressions following the implementation provided by the NLTK library<sup>1</sup>: Below we provide a short description of the “personality” of each agent.

- Eliza<sup>2</sup>: Based on the implementation of Joseph Weizenbaum in 1966 [1], it mimics a non-directional psychotherapist in an initial therapy session.
- Zen<sup>3</sup>: This agent mostly provides philosophical advices following the Zen philosophy.
- Iesha<sup>4</sup>: This is a teen chatbot that discusses about anime and talks in colloquial terms.
- Rude<sup>5</sup>: This agent behaves as a cynical man, giving the user snarky answers

---

<sup>1</sup> <http://www.teachabot.com:8000/main>

<sup>2</sup> [http://nltk.org/\\_modules/nltk/chat/eliza.html](http://nltk.org/_modules/nltk/chat/eliza.html)

<sup>3</sup> [http://nltk.org/\\_modules/nltk/chat/zen.html](http://nltk.org/_modules/nltk/chat/zen.html)

<sup>4</sup> [http://nltk.org/\\_modules/nltk/chat/iesha.html](http://nltk.org/_modules/nltk/chat/iesha.html)

<sup>5</sup> [http://nltk.org/\\_modules/nltk/chat/rude.html](http://nltk.org/_modules/nltk/chat/rude.html)

## 2 Technical Description

The technology behind the aforementioned agents all makes use of simple pattern matching and translation tables to determine the user's intention and to provide a series of answers for each intention. For instance, if the input from the user is "I am a bad person", the agent identifies the word "I am" and captures the rest of the sentence as a group. Then, "I am" is translated as "You are" using a translation table. Thus the result will be "You are a bad person". Next, the system searches for a rule matching a sentence starting with "you are" as in the following example from the agent Rude

```
[r'You are(.*)',  
  ["More like you're %1!",  
   "Hah! Look who's talking.",  
   "Come over here and tell me I'm %1."]]
```

Finally, the chatbot will reply the user picking one answer from the list of options. An interaction using this approach would be:

```
User: You are a bad person  
Rude: Come over here and tell me I'm a bad person
```

When none of the input can be correctly matched in the translation table, the chatbot will reply based on a set of predefined answers to prompt the user for a new topic, or just reply with the given input (in the case of Eliza).

The implementation of these kinds of agents is quick. However, using rule-based techniques conveys some disadvantages and limitations when compared to other conversational agents in the literature, including, but not limited to:

- Vague/invalid responses: Since the answers come from predefined datasets without considering contextual information, and not all inputs can be easily caught using regular expressions, the answers provided by the chatbot may not be contextually correct or what the user wants, as well as they could become predictable and tedious.
- In order to answer every possible user's input, the developer will have to think and predict every possible input creating and maintaining a huge number of patterns. This is the case of A.L.I.C.E. which has won the Loebner prize several times but which requires more than 40K rules [2].

## 3 Sample Dialogue Sessions

Next, we provide some examples of the different answers given for each agent to the same user utterance.

1	<b>User:</b> Tell me more about yourself
	<b>Rude:</b> I'm getting bored here. Become more interesting
	<b>Eliza:</b> how do you feel when you say that?
	<b>Iesha:</b> Ur funny! Kekeke
	<b>Zen:</b> Have you ever considered that you might be a selfish person?
2	<b>User:</b> No
	<b>Rude:</b> Change the subject before I die of fatal boredom
	<b>Eliza:</b> no.
	<b>Iesha:</b> do you watch anime? i like sailor moon ^ ^
	<b>Zen:</b> It is better to be right than to be certain
3	<b>User:</b> Why are you so rude?
	<b>Rude:</b> Why not?
	<b>Eliza:</b> Why don't you tell me the reason why am me so rude?
	<b>Iesha:</b> booooooring!! ur not very fun
	<b>Zen:</b> Are you sure I so rude?
4	<b>User:</b> How old are you?
	<b>Rude:</b> I'm not even going to dignify that with an answer
	<b>Eliza:</b> Perhaps you can answer your own question
	<b>Iesha:</b> booooring! how old r u?
	<b>Zen:</b> Ask yourself not how, but why?
5	<b>User:</b> What is the time now?
	<b>Rude:</b> Do I look like an encyclopedia?
	<b>Eliza:</b> Why do you ask?
	<b>Iesha:</b> man u ask lots of questions
	<b>Zen:</b> seek truth, not what is the time now.

## 4 Skeleton Sample

The released agents include functionalities to communicate with the online platform by using websockets and JSON messages. For instance, if the agent wants to connect to the platform the python code to use is:

```
message = json.dumps({
    "action": "join",
    "key": "API_KEY",
    "name": "Eliza",
    "description": "SHORT DESCRIPTION",
    "image": "eliza.png"
}).encode("utf-8")

self.sendMessage(message)
```

Another type of message that the agents can send is an N-Best list of answers given a user's input. For instance:

```
message = json.dumps({
    "action": "chat",
    "session": "SESSION_ID",
    "payload": {
        "type": "text",
        "text": [
            "Hello, how are you?",
            "Hi, nice to see you again",
            "It is nice to talk to you"
        ]
    }
}).encode("utf-8")

self.sendMessage(message)
```

In general, only the first answer will be presented to the user, but the agent can send several options for annotation purposes. For other messages and additional information about WebChat please refer to [3] and WebChat's dev page<sup>6</sup>.

## 5 Future Work

Currently, we are planning to extend these agents skeletons to connect them to different social messaging platforms such as Telegram or Facebook Messenger. This will increase the number of potential users, and the possibility of collecting and annotating more data that will be available for the research community through WebChat.

## 6 References

1. Weizenbaum, Joseph. "ELIZA—a computer program for the study of natural language communication between man and machine." *Communications of the ACM* 9.1 (1966): 36-45., doi:10.1145/365153.365168.
2. Wallace, Richard S. "The anatomy of ALICE." *Parsing the Turing Test*. Springer Netherlands, 2009. 181-210.
3. Lue Lin, Luis Fdo. D'Haro, and Rafael Banchs. A Web-based Platform for Collection of Human-Chatbot Interactions, Paper accepted in HAI 2016 to appear in Oct. 2016.

---

<sup>6</sup> <http://teachabot.com:8000/dev>