BUILDING CHATBOTS IN PYTHON

Introduction to conversational software

Alan Nichol
Co-founder and CTO, Rasa

A short history

Conversational software is not a new idea!

- Dates back to at least 1960s
- First wave: command line applications
- ELIZA: 1966

Course content

- Implementing smalltalk, ELIZA style
- How to use regex and ML to extract meaning from free-form text
- Build chatbots that can
  - Query a database
  - Plan a trip
  - Help you order coffee
- Handling statefulness

EchoBot I

```
In [1]: USER: Hello!
Out[1]: BOT: I can hear you, you said: 'Hello!'

In [2]: USER: How are you?
Out[2]: BOT: I can hear you, you said: 'How are you?'
```

EchoBot II

```
In [1]: def respond(message):
   :        return "I can hear you! you said: {}".format(message)

In [2]: def send_message(message):
   :        # calls respond() to get response

In [3]: send_message("hello!")
Out[3]: USER: hello!
   ...: BOT : I can hear you! you said: hello!
```

EchoBot III

```
In [1]: import time

In [2]: time.sleep(0.5)
```

BUILDING CHATBOTS IN PYTHON

Let's practice!

BUILDING CHATBOTS IN PYTHON

Creating a personality

Alan Nichol
Co-founder and CTO, Rasa

Why personality?

- Difference between a command line app and a chatbot
- Makes chatbots and voice assistants more accessible and fun to use
- Your users will expect it!

Smalltalk

```
In [1]: responses = {
      :     "what's your name?": "my name is EchoBot",
      :     "what's the weather today?": "it's sunny!"
      : }

In [2]: def respond(message):
      :     if message in responses:
      :         return responses[message]

In [3]: respond("what's your name?")
Out[3]: 'my name is EchoBot'
```

Including variables

```
In [1]: responses = {
      :     "what's today's weather?": "it's {} today"
      : }

In [2]: weather_today = "cloudy"

In [3]: def respond(message):
      :     if message in responses:
      :         return responses[message].format(weather_today)
      :

In [4]: respond("what's today's weather?")
Out[4]: "it's cloudy today"
```

Choosing responses

```
In [1]: responses = {
      :     "what's your name?": [
      :       "my name is EchoBot",
      :       "they call me EchoBot",
      :       "the name's Bot, Echo Bot"
      :     ]
      : }

In [2]: import random

In [3]: def respond(message):
      :     if message in responses:
      :         return random.choice(responses[message])

In [4]: respond("what's your name?")
Out[4]: "the name's Bot, Echo Bot"
```

Asking questions

```
In [1]: responses = [ "tell me more!", "why do you think that?" ]

In [2]: import random

In [3]: def respond(message):
   ...:     return random.choice(responses)

In [4]: respond("I think you're really great")
Out[4]: 'why do you think that?'
```

BUILDING CHATBOTS IN PYTHON

Let's practice!

BUILDING CHATBOTS IN PYTHON

Text processing with regular expressions

Alan Nichol
Co-founder and CTO, Rasa

Regular expressions

- Match messages against known patterns
- Extract key phrases
- Transform sentences grammatically

The regex behind ELIZA

USER: *"do you remember when you ate strawberries in the garden?"*

ELIZA: *"How could I forget when I ate strawberries in the garden?"*

Pattern matching

```
In [1]: import re

In [2]: pattern = "do you remember .*"

In [3]: message = "do you remember when you ate strawberries in the garden"

In [4]: match = re.search(pattern, message)

In [5]: if match:
   ...:     print("string matches!")
Out[5]: string matches!
```

Extracting key phrases

```
In [1]: import re

In [2]: pattern = "if (.*)"

In [3]: message = "what would happen if bots took over the world"

In [4]: match = re.search(pattern, message)

In [5]: match.group(0)
Out[5]: 'what would happen if bots took over the world'

In [6]: match.group(1)
Out[6]: 'bots took over the world'
```

Grammatical transformation

```
In [1]: import re

In [2]: def swap_pronouns(phrase):
   ...:     if 'I' in phrase:
   ...:         return re.sub('I', 'you', phrase)
   ...:     if 'my' in phrase:
   ...:         return re.sub('my', 'your', phrase)
   ...:
   ...:     else:
   ...:         return phrase

In [3]: swap_pronouns("I walk my dog")
Out[3]: 'You walk your dog'
```

Putting it all together

```
In [1]: pattern = 'do you remember (.*)'

In [2]: message = 'do you remember when you ate strawberries in the garden'

In [3]: phrase = pattern.search(pattern, message).group(1)

In [4]: phrase
Out[4]: 'when you ate strawberries in the garden'

In [5]: response = choose_response(pattern)

In [6]: response
Out[6]: 'how could I forget {}'

In [7]: phrase = swap_pronouns(phrase)

In [8]: phrase
Out[8]: 'when I ate strawberries in the garden'

In [9]: response.format(phrase)
Out[9]: 'how could I forget when I ate strawberries in the garden'
```

BUILDING CHATBOTS IN PYTHON

Let's practice!