



BUILDING CHATBOTS IN PYTHON

Stateful bots

Alan Nichol

Co-founder and CTO, Rasa



What do we mean by stateful?

"I love stateless systems!"

"don't they have drawbacks?"

"don't *what* have drawbacks?"



State machines



- Browsing
- Providing address, billing info
- Order complete

Implementing a state machine

```
INIT = 0
CHOOSE_COFFEE = 1
ORDERED = 2
```

Example rules:

```
policy_rules = {
    (INIT, "order"): (CHOOSE_COFFEE, "ok, Columbian or Kenyan?"),
    (CHOOSE_COFFEE, "specify_coffee"):
    (ORDERED, "perfect, the beans are on their way!"),
}
```

Using the state machine

```
In [1]: state = INIT

In [2]: def respond(state, message):
...:     (new_state, response) = policy_rules[(state, interpret(message))]
...:     return new_state, response

In [3]: def send_message(state, message):
...:     new_state, response = respond(state, message)
...:     return new_state

In [4]: state = send_message(state, message)
```



BUILDING CHATBOTS IN PYTHON

Let's practice!



BUILDING CHATBOTS IN PYTHON

Asking questions & queuing answers

Alan Nichol

Co-founder and CTO, Rasa



Reusable patterns

"I'd like some Kenyan beans"

"I'm sorry, we're out of those. Shall I order some Brazilian ones for you?"

"Yes please"

"Can I get a box of 200 brown filters"

"I'm sorry, we're out of those, but I can get your some white ones. Should I order those for you?"

"Yes please"



Pending actions

- Policy returns two values: Selected `action` and `pending_action`
- `pending_action` is saved in the outer scope
- If we get a "yes" intent and there is a pending action, we execute it
- If we get a "no" intent, we wipe any pending actions

Pending state transitions

"I'd like to order some coffee"

```
state = INIT
action = "request_auth"
pending_state = AUTHED
```

- Sounds good! I'd love to help you but you'll have to log in first, what's your phone number?

"555-12345"

```
state = AUTHED
action = "acknowledge_auth"
pending_state = None
```

- Perfect! welcome back :)



BUILDING CHATBOTS IN PYTHON

Let's practice!



BUILDING CHATBOTS IN PYTHON

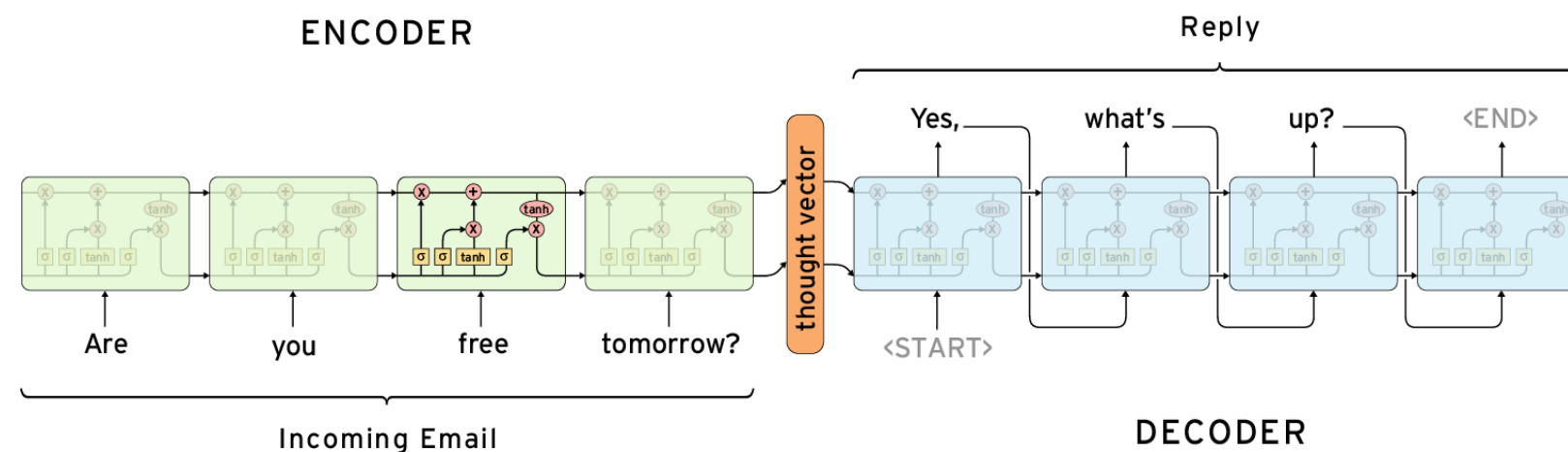
Frontiers of dialogue technology

Alan Nichol

Co-founder and CTO, Rasa



A neural conversational model



"What do you think of Cleopatra?" "Oh, she's very regal"

"What do you think of Messi?" "He's a great player"



Seq2seq

- Machine translation
- Completely data driven, no hand-crafting
- Requires large amount of data
- No guarantee that output is coherent
- Difficult to integrate DB / API calls & other logic

Grounded dialogue systems

- Systems you've built in this course: hand-crafted
- Seq2seq: Data driven
- ML based dialogue systems:
 - NLU
 - Dialogue state manager
 - API logic
 - Natural language response generator
- Human pretend to be a bot: "Wizard of Oz" technique
- Reinforcement learning
 - Receives a reward for a successful conversation



Language generation

- Not recommended if building a bot
- Pre-trained neural network which can generate text
- Scripts of every episode of The Simpsons



Generating sample text

```
generated = sample_text(  
    saved_params,  
    temperature,  
    num_letters=num_letters,  
    init_text=text  
)
```



BUILDING CHATBOTS IN PYTHON

Let's practice!



BUILDING CHATBOTS IN PYTHON

Congratulations!

Alan Nichol

Co-founder and CTO, Rasa



BUILDING CHATBOTS IN PYTHON

Let's practice!