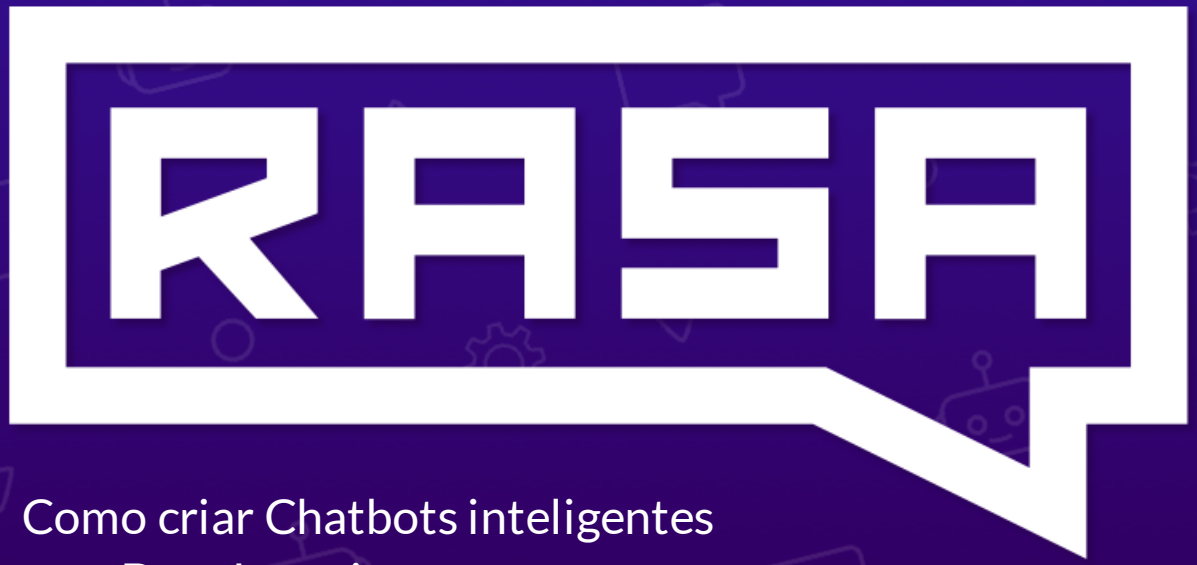


Como criar Chatbots inteligentes
com DeepLearning

ROCKET.CHAT & Rasa.ai

/whoami

ROCKET.CHAT

Mande um e-mail para

diego.dorgam@rocket.chat

@diego.dorgam

ACESSE

https://**open**.rocket.chat

/define chatbots

CATEGORIAS

Process Automation
Productivity
Chatbot
Virtual Assistant

Robotic Process Automation

Also known as RPA, this is a kind of robot made for automate tasks made by humans in a way where the robot interacts with a user interface

This kind of approach is gaining scale at services integration where a API is not possible.

Those are screen scraping robots, that can easily assume a human position in tasks like inputting data into system forms, collecting data from reports and internet, and all kinds of repetitive tasks made by humans.

CATEGORIAS

Process Automation
Productivity
Chatbot
Virtual Assistant

Bots for Productivity

Those are bots created to integrate process and application into your chat environment.

By giving a fast approach to productivity systems, like git repositories, trello, twitter, or even mass internal communication tools, these bots make life easier and much more agile.

One branch of this discussion is the ChatOps culture, one of the DevOps abilities that organizations must develop.

CATEGORIAS

Process Automation
Productivity
Chatbot
Virtual Assistant

ChatBots

Most booming technology on the market!

Chatbots combine natural language understanding with service integration technologies (APIs) to deliver services and information.

Chatbots are getting the attention of the big players, and offering a new paradigm on how customers interact with brands.

CATEGORIAS

Process Automation
Productivity
Chatbot
Virtual Assistant

Virtual Assistants

Those are very specialized Artificial Intelligence bots, that are designed to help humans with some personal tasks.

Organize your schedules, find a restaurant nearby, playing a list of your favourite music, those are some tasks that Virtual Assistants can do.

They are intended to help users with the daily tasks, by making them easier to access through voice commands, for example.

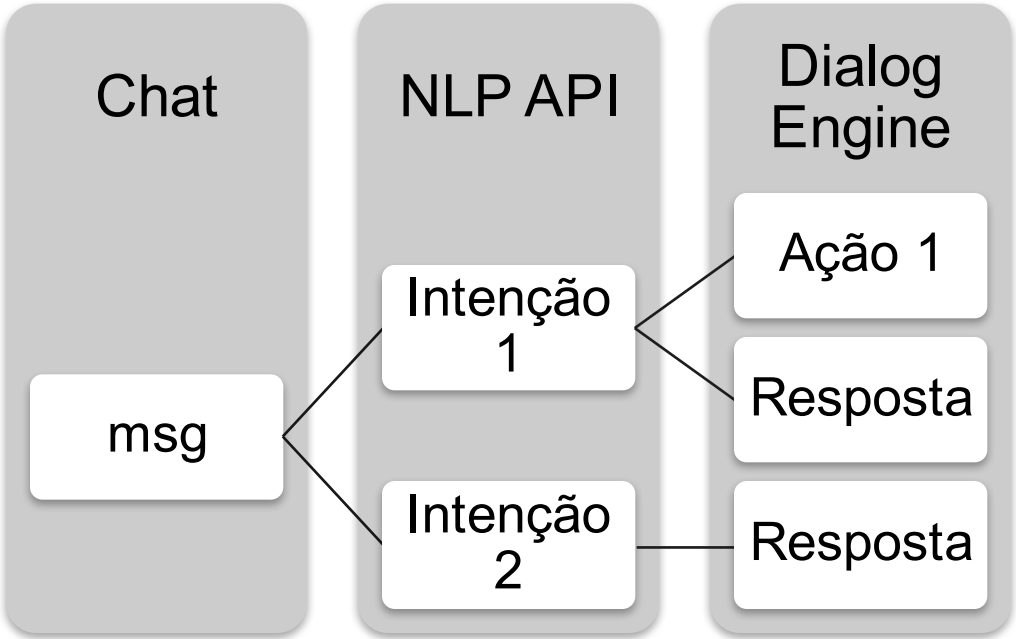
How chatbots work

Bot Framework

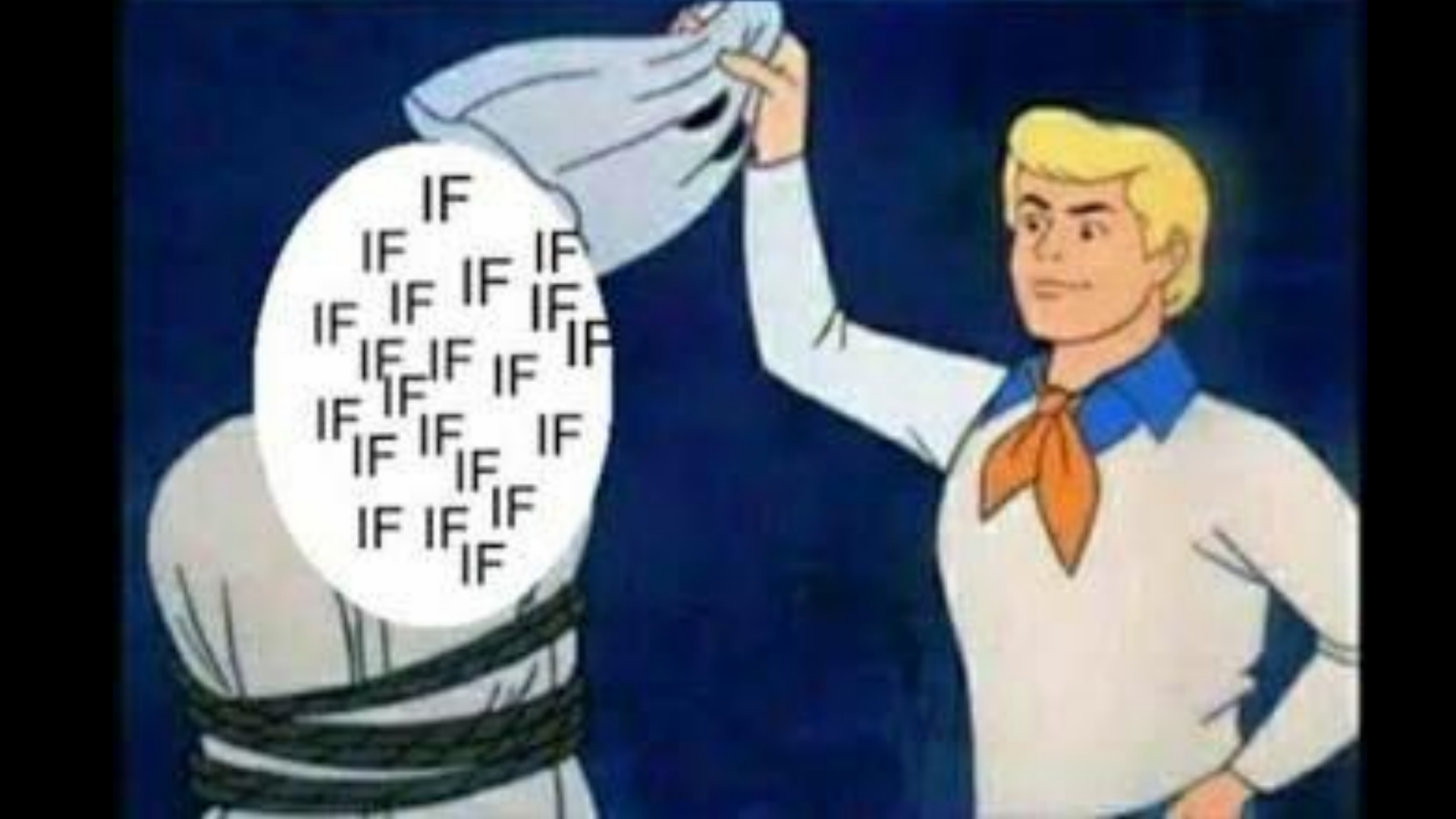
botpress

RedBot

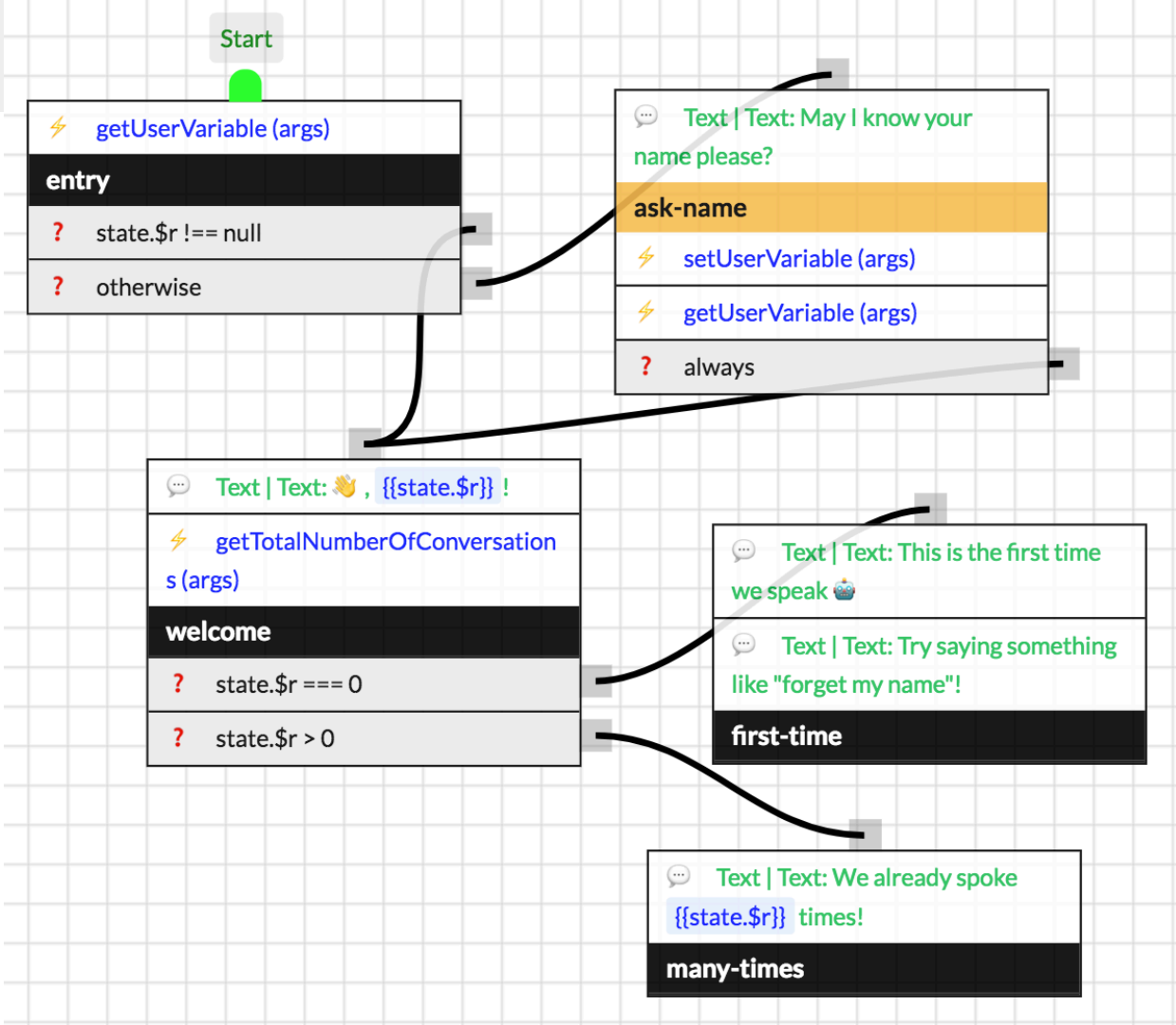
OPSDROID

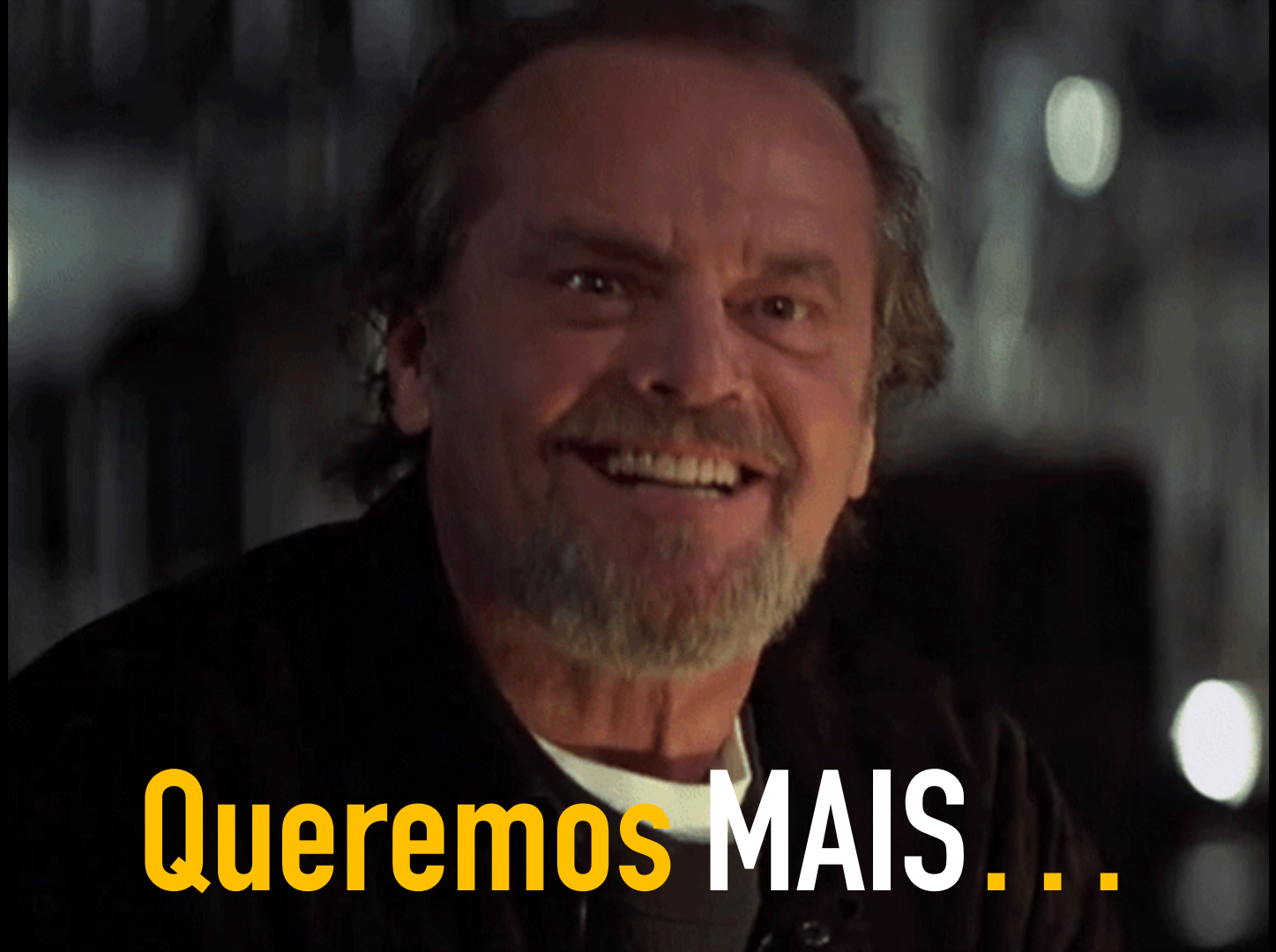



Okay gang, let's see who the advanced AI really is.


IF
IF IF IF
IF IF IF IF IF
IF IF IF IF
IF IF
IF IF IF IF
IF IF IF
IF IF IF
IF

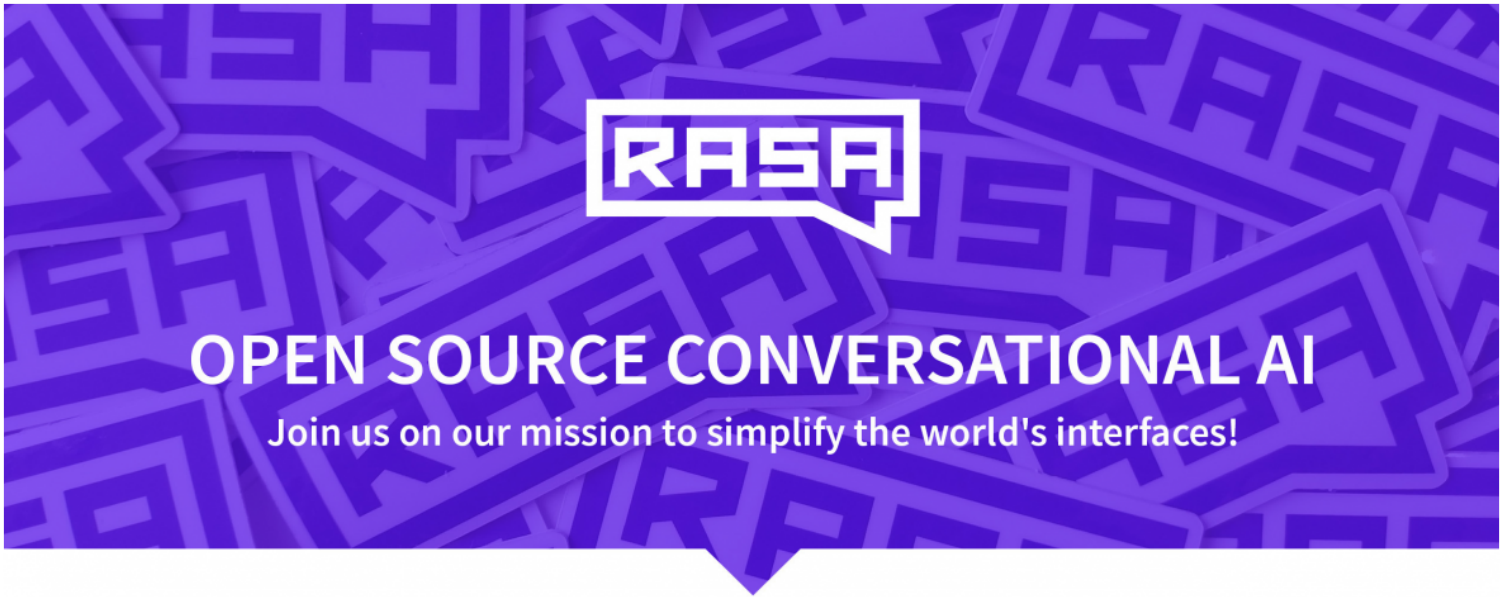
botpress


Start
getUserVariable (args)
entry
state.$r !== null
otherwise
Text | Text: May I know your name please?
ask-name
setUserVariable (args)
getUserVariable (args)
always
Text | Text: 👋 , {{state.$r}} !
getTotalNumberOfConversations (args)
welcome
state.$r === 0
state.$r > 0
Text | Text: This is the first time we speak 🤖
Text | Text: Try saying something like "forget my name"!
first-time
Text | Text: We already spoke {{state.$r}} times!
many-times


Queremos MAIS...

All power of bleeding edge deep learning technologies to build clever bots

Rasa Stack

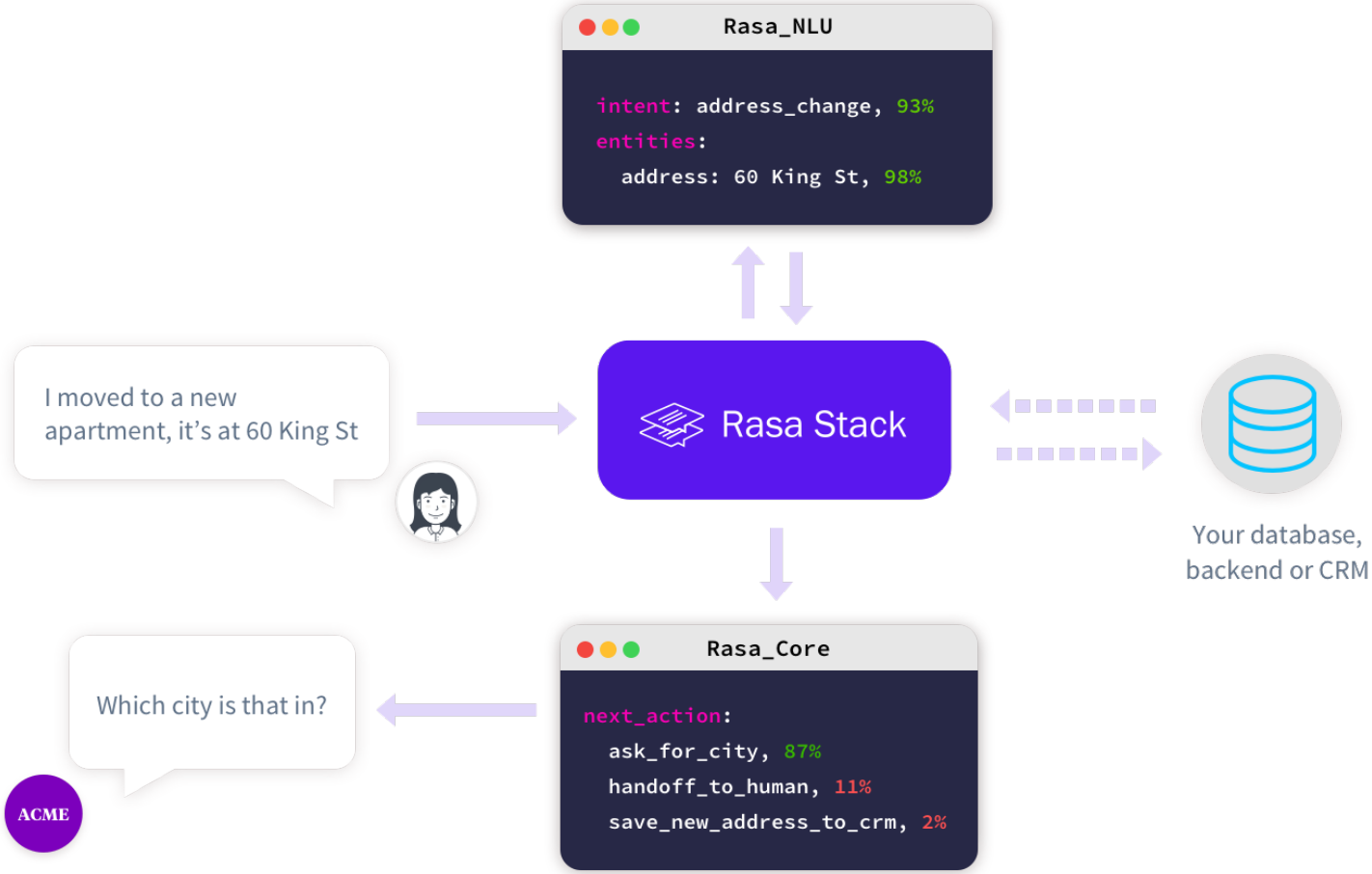

RASA
OPEN SOURCE CONVERSATIONAL AI
Join us on our mission to simplify the world's interfaces!


scikit
learn


TensorFlow


K
Keras


spaCy


Rasa_NLU
intent: address_change, 93%
entities:
address: 60 King St, 98%
I moved to a new apartment, it's at 60 King St
Rasa Stack
Your database, backend or CRM
Rasa_Core
next_action:
ask_for_city, 87%
handoff_to_human, 11%
save_new_address_to_crm, 2%
Which city is that in?
ACME

RASA STACK

ACME

Where should we send the confirmation to?

Please send it to amy@example.com

👍

ACME

Should we make that your primary email?

Oh, what is it right now?

Next best action:

- [x] `fetch_primary_email` 88%
- [ ] `hand_off_to_human` 8%
- [ ] `fetch_primary_phone` 4%

RASA STACK

nlu_model_config.yml

pipeline: "spacy_sklearn"

nlu_model_config.yml

```
pipeline:
   - name: "nlp_spacy"
   - name: "tokenizer_spacy"
   - name: "intent_featurizer_spacy"
   - name: "intent_classifier_sklearn"
   - name: "ner_crf"
   - name: "ner_synonyms"
```

NLU

RASA Natural Language Understanding

Um **interpretador** próprio que pode funcionar como um server http:

"I am looking for Mexican food"

retorna:


```
{
    "intent": {
    "name": "restaurant_search",
    "confidence": 0.8231117999072759
    },
    "entities": [
        {
            "start": 17,
            "end": 24,
            "value": "mexican",
            "entity": "cuisine",
            "extractor": "ner_crf"
        }
    ],
    "intent_ranking": [
        {
            "name": "restaurant_search",
            "confidence": 0.8231117999072759
        },
        {
            "name": "affirm",
            "confidence": 0.07618757211779097
        },
        {
            "name": "goodbye",
            "confidence": 0.06298664363805719
        },
        {
            "name": "greet",
            "confidence": 0.03771398433687609
        }
    ],
    "text": "I am looking for Mexican food"
}
```


```
curl -X POST "localhost:5000/parse" -d "{/"q/":/"I am looking for Mexican food/"}" | python -m json.tool
```

CORE

`intents`	things you expect users to say. See Rasa NLU for details.
`entities`	pieces of info you want to extract from messages. See Rasa NLU for details.
`actions`	things your bot can do and say
`slots`	information to keep track of during a conversation (e.g. a users age)
`templates`	template strings for the things your bot can say

domain.yml

```
intents:
  - greet
  - goodbye
  - mood_affirm
  - mood_deny
  - mood_great
  - mood_unhappy
```

nlu.md

```
## intent:greet
- hey
- Hello
- Hi
- hello there
- good morning
- good evening

## intent:goodbye
- good by
- cee you later
- good night
- good afternoon
- Bye
- Goodbye

## intent:mood_affirm
- Yes
- indeed-
- of course-
- that sounds good-
- Correct

## intent:mood_deny
- No
- Never
- I don't think so
- don't like that
- no way
```

domain.yml

```
actions:
- utter_greet
- utter_cheer_up
- utter_did_that_help
- utter_happy
- utter_goodbye
```

domain.yml

```
templates:
  utter_greet:
  - text: "Hey! How are you?"
    buttons:
    - title: "great"
      payload: "great"
    - title: "super sad"
      payload: "super sad"

  utter_cheer_up:
  - text: "Here is something to cheer you up:"
    image: "https://i.imgur.com/nGF1K8f.jpg"
```

domain.yml

```yaml
intents:
  - greet
  - goodbye
  - mood_affirm
  - mood_deny
  - mood_great
  - mood_unhappy

actions:
- utter_greet
- utter_cheer_up
- utter_did_that_help
- utter_happy
- utter_goodbye
```

```yaml
templates:
  utter_greet:
  - text: "Hey! How are you?"
    buttons:
    - title: "great"
      payload: "great"
    - title: "super sad"
      payload: "super sad"

  utter_cheer_up:
  - text: "Here is something to cheer you up:"
    image: "https://i.imgur.com/nGF1K8f.jpg"

  utter_did_that_help:
  - text: "Did that help you?"

  utter_happy:
  - text: "Great carry on!"

  utter_goodbye:
  - text: "Bye"
```

stories.md

```
## happy path
* greet
   - utter_greet
* mood_great
   - utter_happy

## sad path 1
* greet
   - utter_greet
* mood_unhappy
   - utter_cheer_up
   - utter_did_that_help
* mood_affirm
   - utter_happy

## sad path 2
* greet
   - utter_greet
* mood_unhappy
   - utter_cheer_up
   - utter_did_that_help
* mood_deny
   - utter_goodbye

## say goodbye
* goodbye
   - utter_goodbye
```

Interactive Learning

```
/greet
------
Chat history:

     bot did:    None
     bot did:        action_listen
     user said:      /greet

              whose intent is:        greet

we currently have slots: concerts: None, venues: None
------
The bot wants to [utter_greet] due to the intent. Is this correct?

    1.       Yes
    2.       No, intent is right but the action is wrong
    3.       The intent is wrong
    0.       Export current conversations as stories and quit
```


“
UNIVERSAL LAW IS FOR LACKEYS. CONTEXT IS FOR KINGS.
GABRIEL LORCA
MAGICALQUOTE
WWW.MAGICALQUOTE.COM

custom policy

```python
from rasa_core.policies import Policy
from rasa_core.actions.action import ACTION_LISTEN_NAME
from rasa_core import utils
import numpy as np

class SimplePolicy(Policy):
    def predict_action_probabilities(self, tracker, domain):
        responses = {"greet": 3}

        if tracker.latest_action_name == ACTION_LISTEN_NAME:
            key = tracker.latest_message.intent["name"]
            action = responses[key] if key in responses else 2
            return utils.one_hot(action, domain.num_actions)
        else:
            return np.zeros(domain.num_actions)
```

custom policy

In pseudocode, what the `SimplePolicy` above does is:

```
-> a new message has come in

if we were previously listening:
    return a canned response
else:
    we must have just said something, so let's Listen again
```


TALK IS CHEAP
SHOW ME THE CODE
memegenerator.net


TALK IS CHEAP
SHOW ME THE CODE

https://github.com/lappis-unb/rouana/

/policies

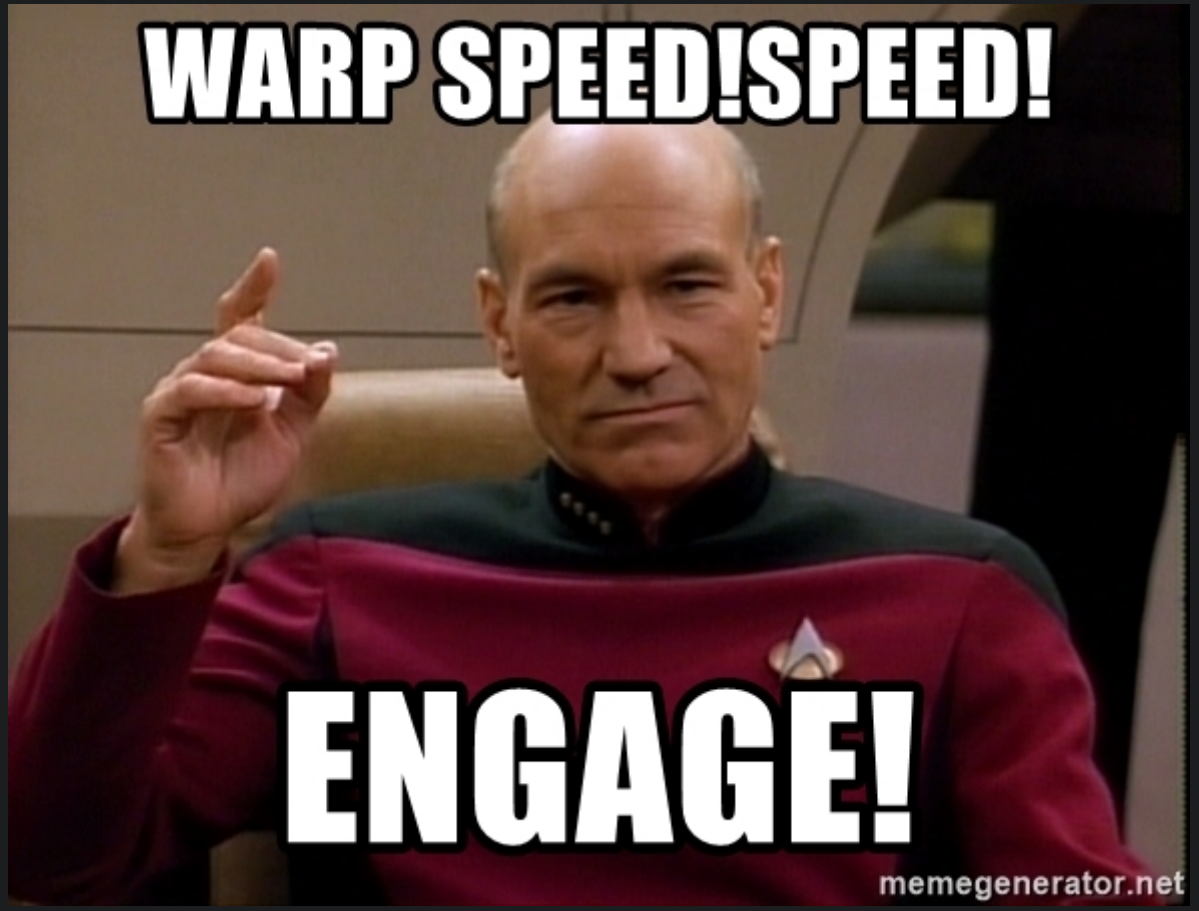
```
def train_dialogue(domain_file='domain.yml',
                   model_path='models/dialogue',
                   training_data_file='data/stories'):
    fallback = FallbackPolicy(fallback_action_name="utter_default",
                              core_threshold=0.11,
                              nlu_threshold=0.11)

    agent = Agent(
        domain_file,
        policies=[KerasPolicy(), MemoizationPolicy(max_history=4), fallback]
    )


    training_data = agent.load_data(training_data_file)
    agent.train(
        training_data,
        epochs=TRAINING_EPOCHS,
        batch_size=256,
        validation_split=0.20
    )

    agent.persist(model_path)
    return agent
```

/policies

/policies

```
9  def train_dialogue(domain_file, model_path, training_folder):
10
11     agent = Agent(domain_file, policies=[
12             MemoizationPolicy(max_history=6),
13             KerasPolicy(MaxHistoryTrackerFeaturizer(BinarySingleStateFeaturizer(),
14                                                     max_history=6)),
15             FallbackPolicy(nlu_threshold=0.8, core_threshold=0.3)])
16
17     training_data = agent.load_data(training_folder)
18
19     agent.train(training_data, epochs=100)
20     agent.persist(model_path)
```

/demo



Dúvidas?

Experience the next level of team chat platform.

Simple, free, for you.

Mande um e-mail para

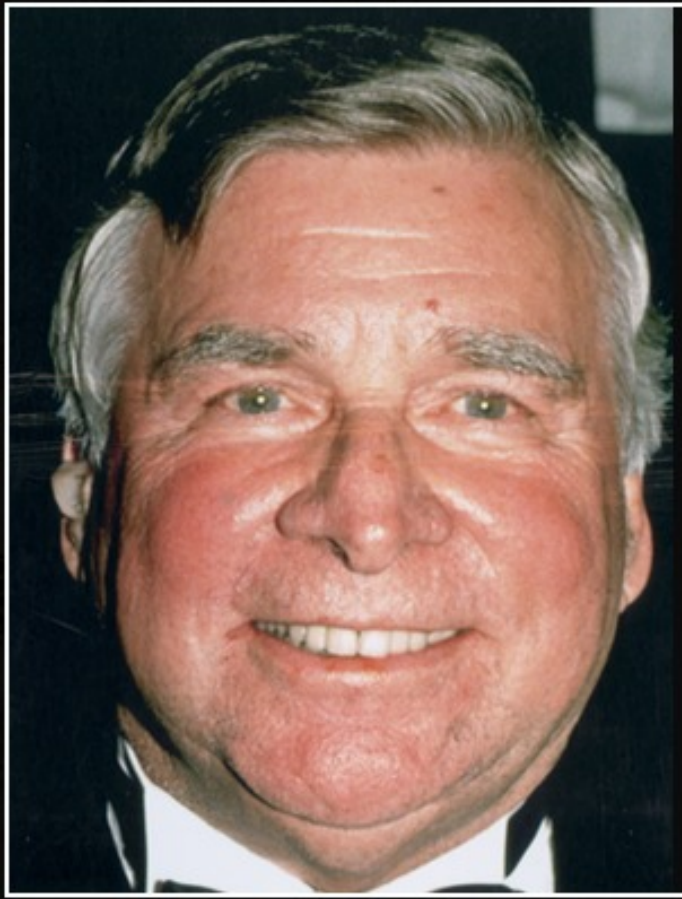
diego.dorgam@rocket.chat

@diegodorgam

Converse com a gente no Rocket.Chat

Acesse o servidor do time da Rocket.Chat no link open.rocket.chat . Teste a plataforma e nos encontro nos canais:

- #general General discussion channel
- #support Rocket.Chat support channel



Star Trek was an attempt to say that humanity will reach maturity and wisdom on the day that it begins not just to tolerate, but take a special delight in differences in ideas and differences in life forms.

— Gene Roddenberry —


PRBK
#ELE
NÃO