CHATOPS 101

WITH OPSDROID

WHAT THE F#!@ IS CHATOPS?

> *ChatOps is the use of chat clients, chat-bots and real-time communication tools to facilitate how software development and operation tasks are communicated and executed.*

Abhinav Jain, works at Accenture and sometimes answers questions like this on Quora.

SO HERE IS OPSDROID

Opsdroid is an open source ChatOps bot framework with the moto: **Automate boring things!**

BUT WHY OPSDROID?

BUT WHY OPSDROID?

Simple: Easy to install, configure and deploy.

BUT WHY OPSDROID?

Simple: Easy to install, configure and deploy.

Powerful: Works out of the box with Slack, Telegram, Facebook… and with various NLU platforms.

BUT WHY OPSDROID?

Simple: Easy to install, configure and deploy.

Powerful: Works out of the box with Slack, Telegram, Facebook… and with various NLU platforms.

Extensible: Add your custom skills in few Python lines.

SHOW ME MORE

LET'S SEE HOW OPSDROID WORKS

SKILLS

Skills are modules which define what actions opsdroid should perform based on different chat messages.

They’re modular and can be shared as plugins between differents opsdroid instances.

SKILLS

```python
class HelloSkill(Skill):

  @match_regex(r'hi|hello|hey|hallo')
  async def hello(self, message: Message):
    text = random.choice(
      ["Hi {}", "Hello {}", "Hey {}"]
    ).format(message.user)
    await message.respond(text)

  @match_regex(r'bye( bye)?|see y(a|ou)|au revoir|I(\')?M off')
  async def goodbye(self, message: Message):
    text = random.choice(
      ["Bye {}", "See you {}", "Au revoir {}"]
    ).format(message.user)
    await message.respond(text)
```

PARSERS

Parsers match an incoming message to a skill.

Actual parsers: *Regex*, *Parse_Format*, *Crontab*, *Webhook*, *Always and NLU parsers*

PARSERS

PARSERS

```python
class MyNameSkill(Skill):

  @match_regex(r'my name is (?P<name>\w+)')
  async def my_name_is(self, message: Message):
    name = message.regex.group('name')
    await message.respond(f'Wow, {name} is a nice name!')
```

PARSERS

```python
class MyNameSkill(Skill):

  @match_regex(r'my name is (?P<name>\w+)')
  async def my_name_is(self, message: Message):
    name = message.regex.group('name')
    await message.respond(f'Wow, {name} is a nice name!')
```

```python
class MyNameSkill(Skill):

  @match_parse('my name is {name}')
  async def my_name_is(self, message: Message):
    name = message.parse_result['name']
    await message.respond(f'Wow, {name} is a nice name!')
```

PARSERS

```python
class ClockSkill(Skill):

    @match_crontab('0 * * * *')
    @match_regex(r'what time is it\?')
    async def speaking_clock(self, message: Message):
      connector = self.opsdroid.default_connector
      default_room = connector.default_room

      if message is None:
        message = Message('', None, default_room, connector)

      await message.respond(strftime("It's %H:%M", gmtime()))
```

CONNECTORS

Connectors are modules for connecting opsdroid to your specific chat service.

Actual connectors: *Shell*, *Websocket*, *Slack*, *Telegram*, *Twitter*, *Facebook*, *Github*, *Ciscospark and Matrix*

CONFIG

```yaml
parsers:
  - name: witai
    enabled: true
    access-token: "mysecretwittoken"
    min-score: 0.7

connectors:
  - name: slack
    token: "mysecretslacktoken"

skills:
  - name: hello
  - name: myawesomeskill
    repo: "https://github.com/username/myawesoneskill.git"
```

YEAH, BUT...

YOU SAID SOMETHING ABOUT NLU?

WHAT THE HECK IS NLU?

Natural language understanding (NLU) is a branch of artificial intelligence (AI) that uses computer software to understand input made in the form of sentences in text or speech format.

Margaret Rouse in WhatIs.com

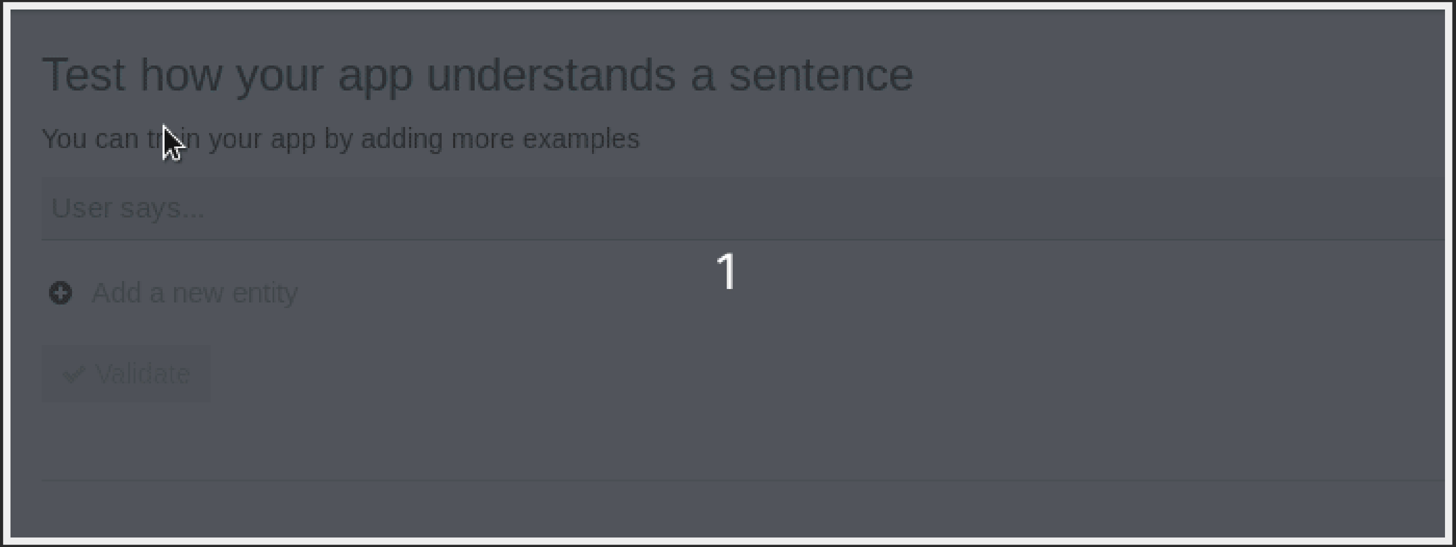
NLU PARSERS

Opsdroid connects with some NLU services:

- **Wit.ai** (Facebook service)
- **Dialogflow** (Google service)
- **Luis.AI** (Microsoft service)
- **Recast.AI** (SAP service)
- **Rasa** (Open Source)

WIT.AI EXAMPLE

WIT.AI RESTART EXAMPLE

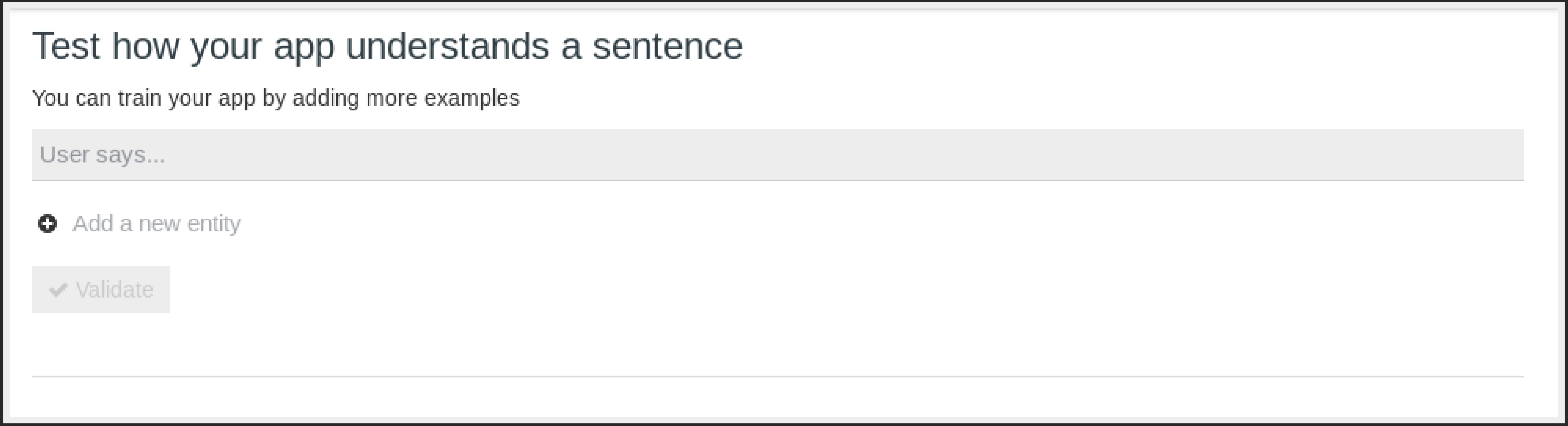

Test how your app understands a sentence

You can train your app by adding more examples

User says...

1

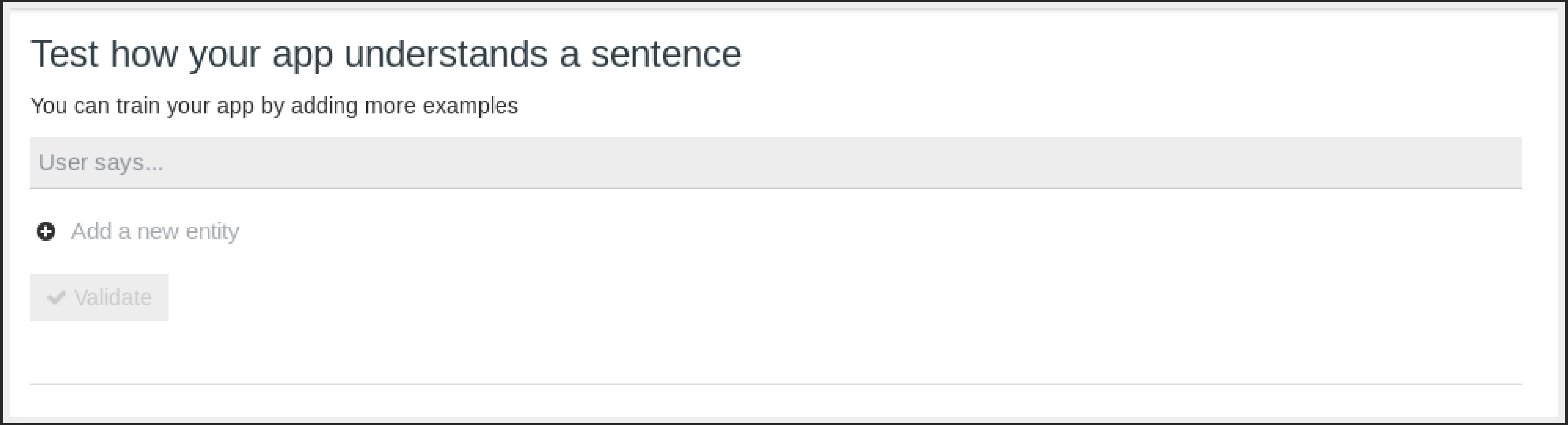
WIT.AI RESTART EXAMPLE



Test how your app understands a sentence

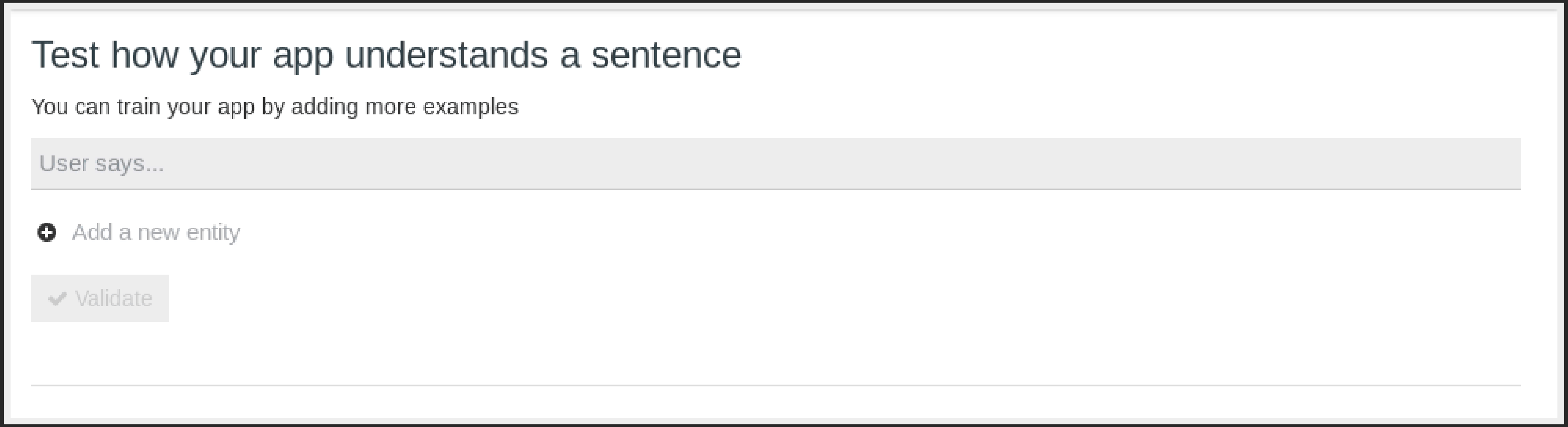
You can train your app by adding more examples

WIT.AI RESTART EXAMPLE


Test how your app understands a sentence

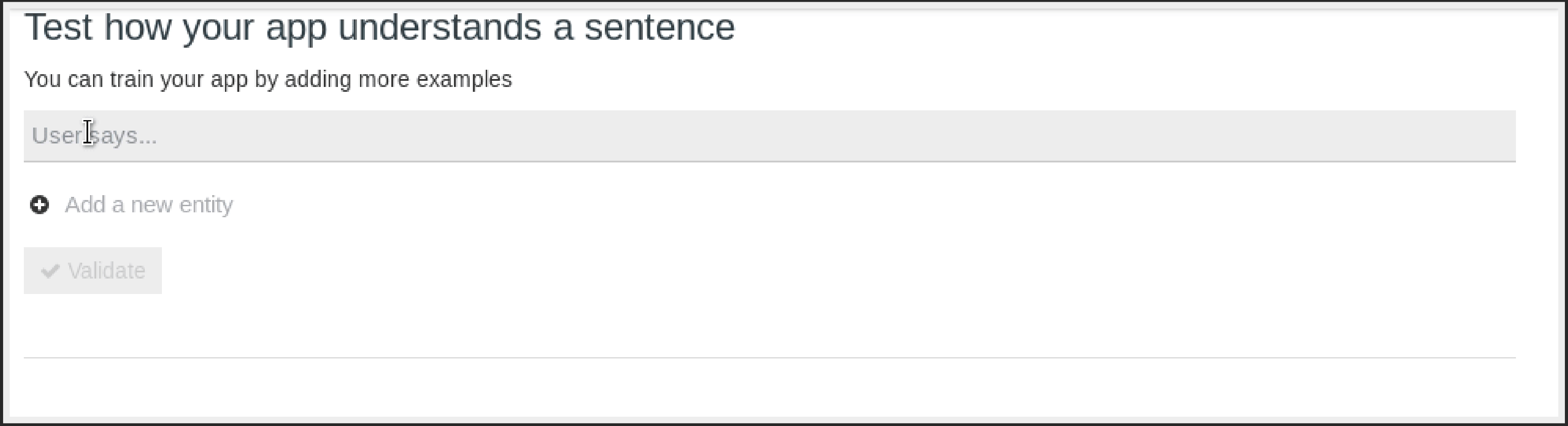
You can train your app by adding more examples

WIT.AI RESTART EXAMPLE


Test how your app understands a sentence

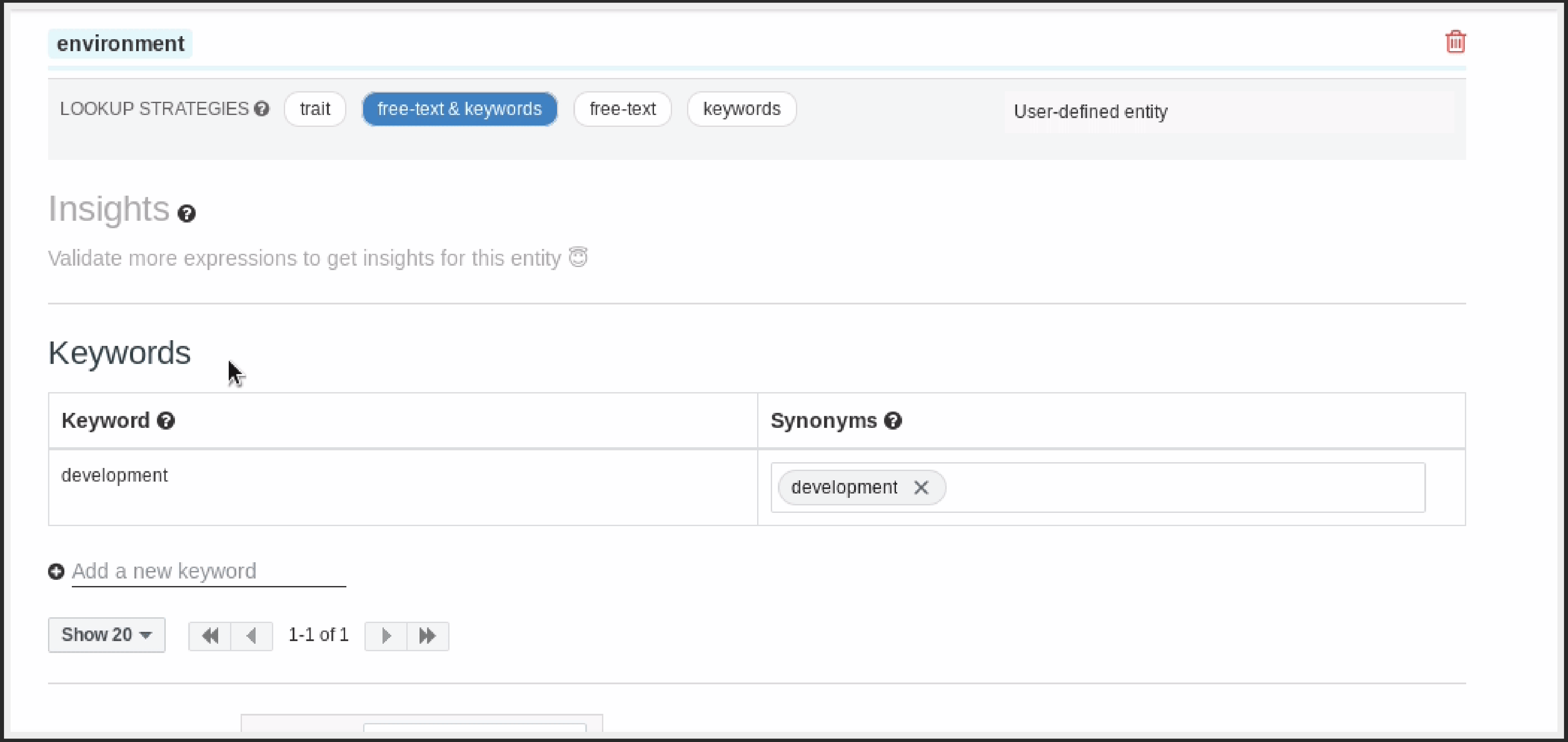
You can train your app by adding more examples

WIT.AI RESTART EXAMPLE


Test how your app understands a sentence

You can train your app by adding more examples

WIT.AI RESTART EXAMPLE



environment

LOOKUP STRATEGIES: trait, [✓ free-text & keywords], free-text, keywords

User-defined entity

Insights

Validate more expressions to get insights for this entity

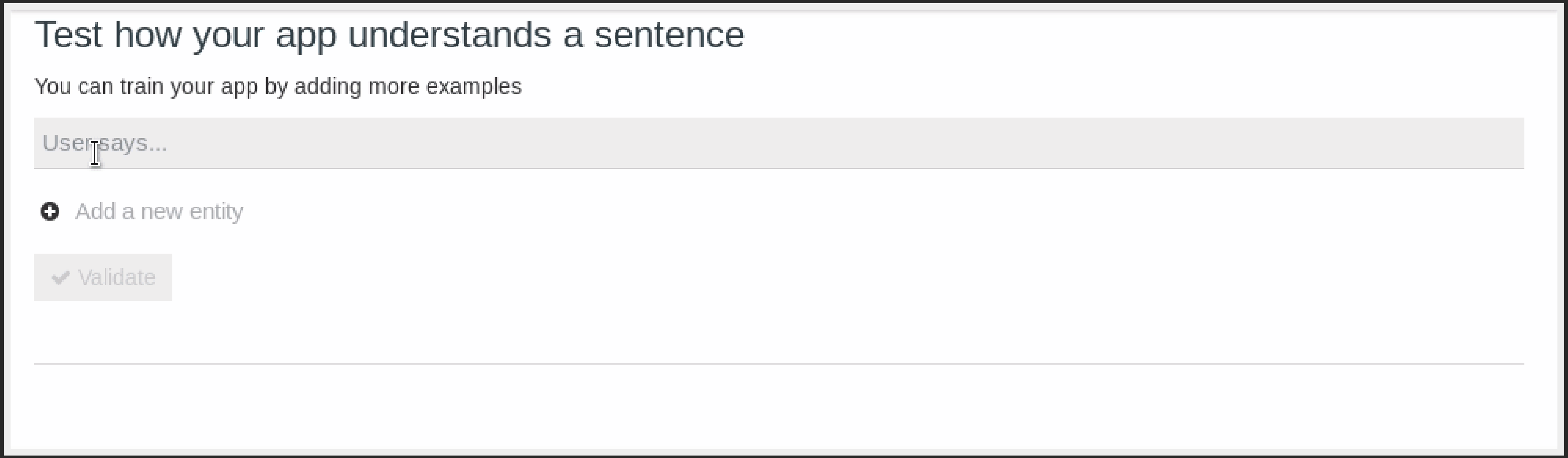
Keywords

Keyword	Synonyms
development	development

Add a new keyword

Show 20 · 1-1 of 1

WIT.AI RESTART EXAMPLE


Test how your app understands a sentence

You can train your app by adding more examples

WIT.AI API EXAMPLE

A message "*restart production, please!*" is sent to Wit.ai

```
{
  "_text": "restart production, please!",
  "entities": {
    "intent": [
      {
        "confidence": 0.98,
        "value": "restart"
      }
    ],
    "environment": [
      {
        "confidence": 1,
        "value": "production",
        "type": "value"
      }
    ]
```

WIT.AI CODE EXAMPLE

```
class RestartSkill(Skill):

  @match_witai('restart')
  async def restart(self, message: Message):
    entities = message.witai['entities']
    environments = entities['environment']
    if not environments:
      await message.respond('Please specify an environment.')
      return

    environment = environments['0']['value']
    await _do_restart(environment)
    await message.respond(f'{environment} restarted!')
```

SOUNDS COOL, DOESN'T IT?

THANKS! 🤗

ANY QUESTIONS?

I am Àngel, a.k.a. @anxodio

Python Developer / Data Engineer at @HolaluzEng

holaluz is looking for great people like you, join us! holaluz.com/jobs