Resources

Ipython Notebook we are going to start with:

http://tiny.cc/rasa-tut

All files for this workshop:

https://github.com/tmbo/rasa-demo-pydata18

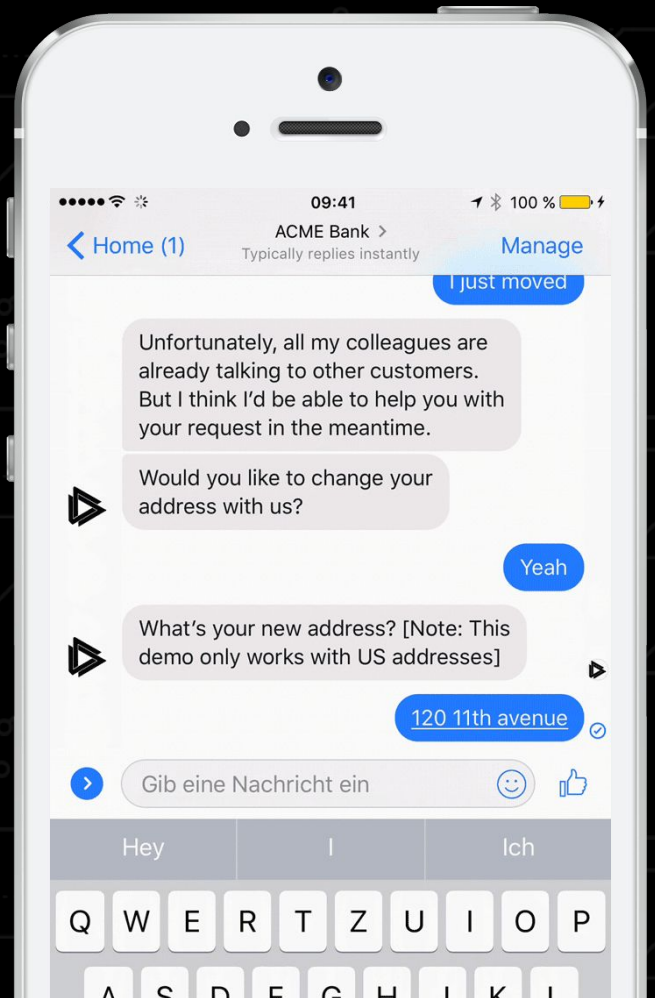
You'll need Ipython and python (preferrably 3.6).

RASA

Conversational AI with Rasa NLU & Rasa Core

Tom Bocklisch
Head of Engineering

Conversational AI will <u>dramatically</u> change how your users interact with you.

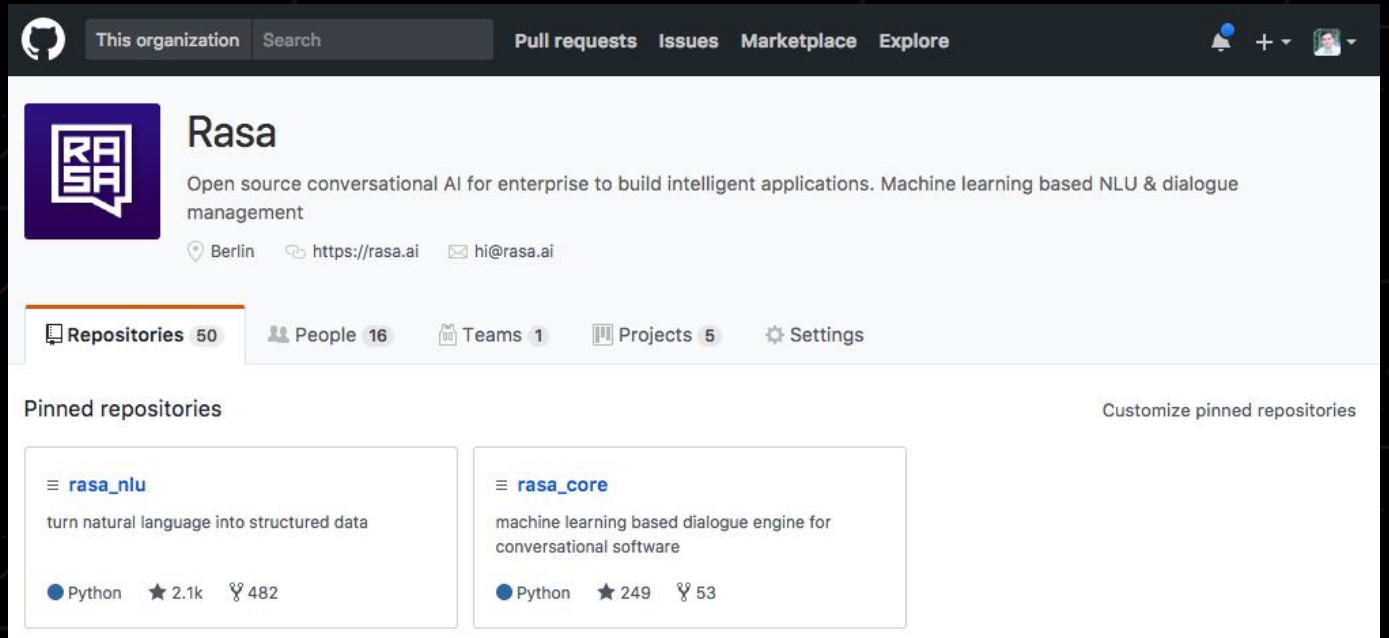


Example:
A customer can change her address via Facebook Messenger

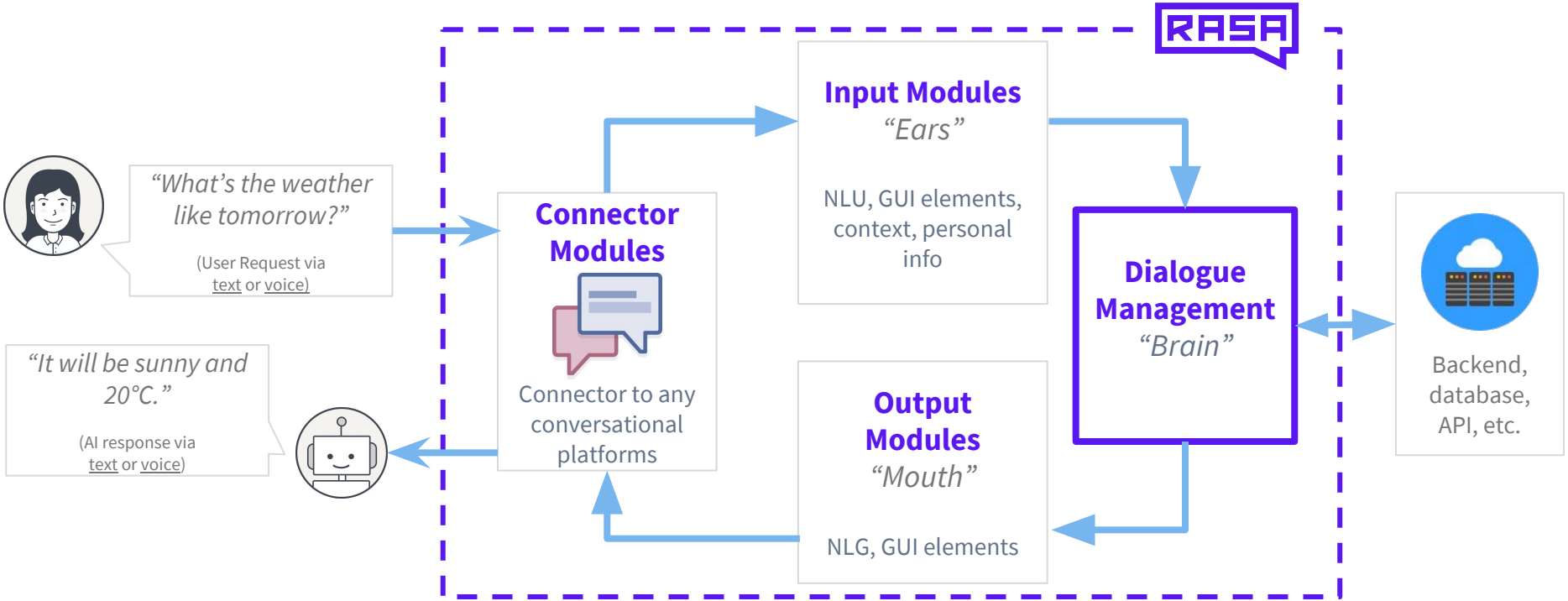
Play demo video

RASA

An open source, highly scalable ML framework to build conversational software



Rasa the OSS to build conversational software with ML



Alternatives: Dialogflow wit.ai

Why Rasa?

Runs Locally

- No Network Overhead
- Control QoS
- Deploy anywhere

Own Your Data

- Don’t hand data over to big tech co’s
- Avoid vendor lock-in

Hackable

- Tune models for your use case

About Me

- Studied Computer Science in Germany
- Went on to found a consultancy focused on image and text analysis - https://scalableminds.com
- Joined the Rasa Team http://rasa.com - leading the development of the open source stack as well as internal tools

If you have any questions about this talk, Rasa or me - I'll be around.

What we are focusing on today

Goal:

build & understand a bot based on machine learning

Roadmap:

1. Natural Language Understanding
    i. Theory
    ii. Let’s Code
2. Dialogue Handling
    i. Theory
    ii. Let’s Code
3. Research
4. Questions

Setup

1. Jupyther notebook in python 3.6 (2.7 should work as well)

```
➜ pydata-18-AMS  mkvirtualenv --python=/usr/local/bin/python3 pydata3
Running virtualenv with interpreter /usr/local/bin/python3
Using base prefix '/usr/local/Cellar/python/3.6.5/Frameworks/Python.framework/Versions/3.6'
New python executable in /Users/tmbo/.virtualenvs/pydata3/bin/python3.6
Also creating executable in /Users/tmbo/.virtualenvs/pydata3/bin/python
Installing setuptools, pip, wheel...done.
virtualenvwrapper.user_scripts creating /Users/tmbo/.virtualenvs/pydata3/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/tmbo/.virtualenvs/pydata3/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/tmbo/.virtualenvs/pydata3/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/tmbo/.virtualenvs/pydata3/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/tmbo/.virtualenvs/pydata3/bin/get_env_details
(pydata3) ➜ pydata-18-AMS  pip install jupyter
Collecting jupyter
```

2. Download:

Ipython Notebook: http://tiny.cc/rasa-tut

Repository: https://github.com/tmbo/rasa-demo-pydata18

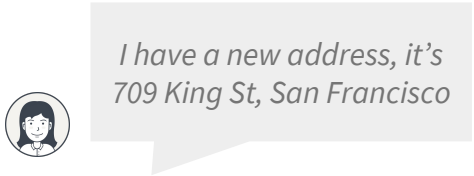
Under The Hood

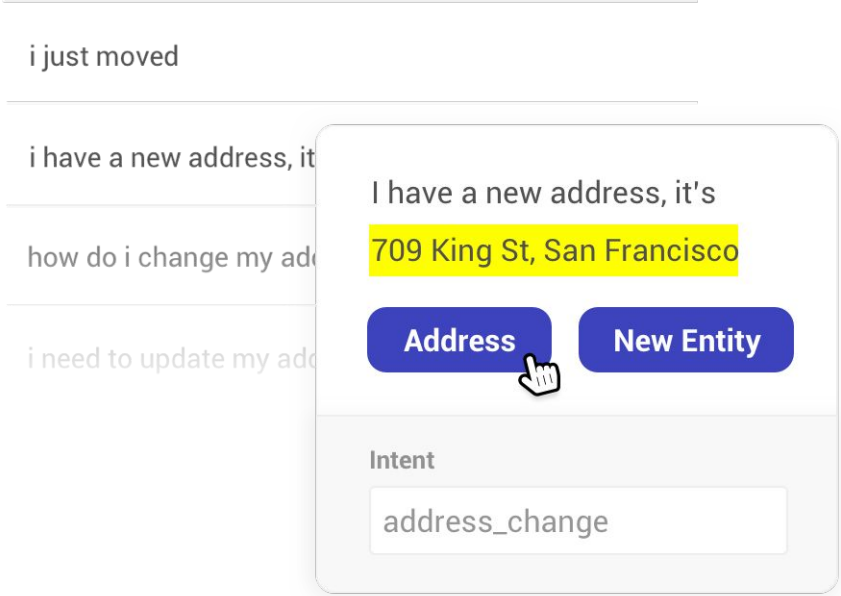
Natural Language Understanding

RASA

Rasa NLU: Natural Language Understanding

Goal: create structured data





Rasa NLU: Natural Language Understanding

What's the weather like tomorrow?

Natural Language Understanding

Example Intent Classification Pipeline

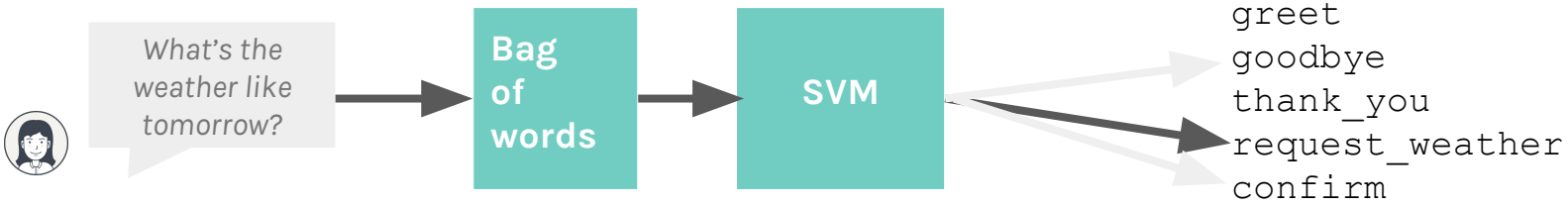
"What's the weather like tomorrow?" { "intent": "**request_weather**" }

Vectorization → Intent Classification

Rasa NLU: Natural Language Understanding

Bags are your friend

$$\{v_1, \ldots, v_s\} \rightarrow \frac{1}{s} \sum_i v_i$$



Rasa NLU: Natural Language Understanding

What's the weather like tomorrow?

Natural Language Understanding

Example Intent Classification Pipeline

"What's the weather like tomorrow?" { "intent": "**request_weather**" }

Vectorization → Intent Classification

Example Entity Extraction Pipeline

"What's the weather like **tomorrow**?"

{ "date": "**tomorrow**" }

Tokenizer → Part of Speech Tagger → Chunker → Named Entity Recognition → Entity Extraction

Rasa NLU: Entity Extraction

Where can I get a burrito in the 2nd arrondissement ?

cuisine

location

averaged perceptron

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{K} c^{(k)}\left(w^{(k)} \cdot \hat{x} + b^{(k)}\right)\right)$$

1. Binary classifier `is_entity` & then `entity_classifier`
2. Direct structured prediction

Let’s Code

Natural Language Understanding

RASA

Let’s build a fun little bot

hello

Hey! How are you?

great

super sad

super sad

Here is something to cheer you up:

Did that help you?

Let’s build a fun little bot

jupyter rasa-moodbot-demo-starter Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Markdown

Conversational AI with Rasa

This notebook is going to be the basis for my workshop at the PyData 2018 Amsterdam workshop. If you have any questions, please let me know!

You'll build a relatively simple bot, that just asks you about your mood and tries to cheer you up if you're feeling a bit down.

The tutorial consists of three parts:

- Part 0: Installation and preparations
- Part 1: You'll start with a basic bot that can only understand natural language but no dialogues.
- Part 2: You'll add the abilitiy to understand multiturn dialogues.
- Part 3: I'll give you further resources so you can extend this simple demo.

Under The Hood

Dialogue Handling

Let’s get you awake again!

Everyone get a partner and an arms length of space.

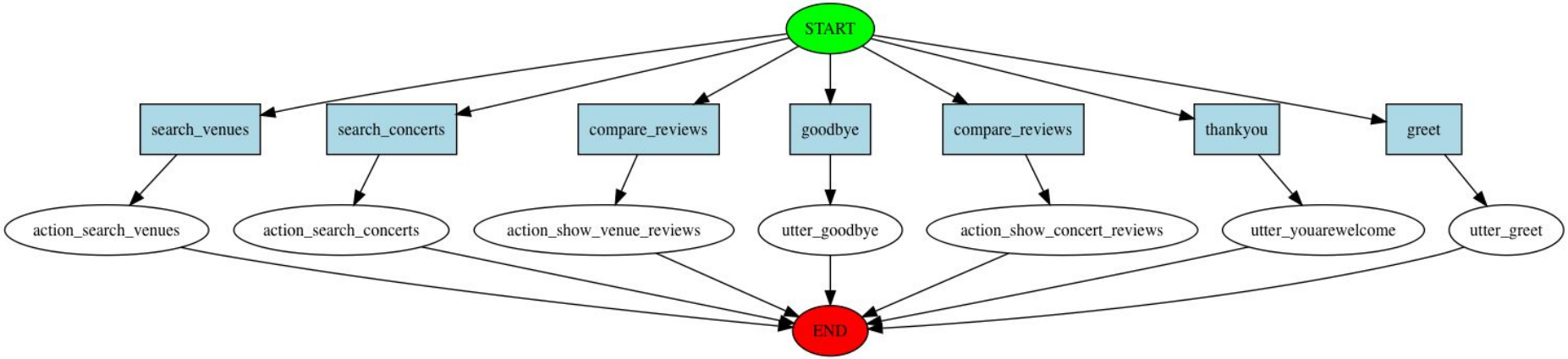
Under The Hood

Dialogue Handling

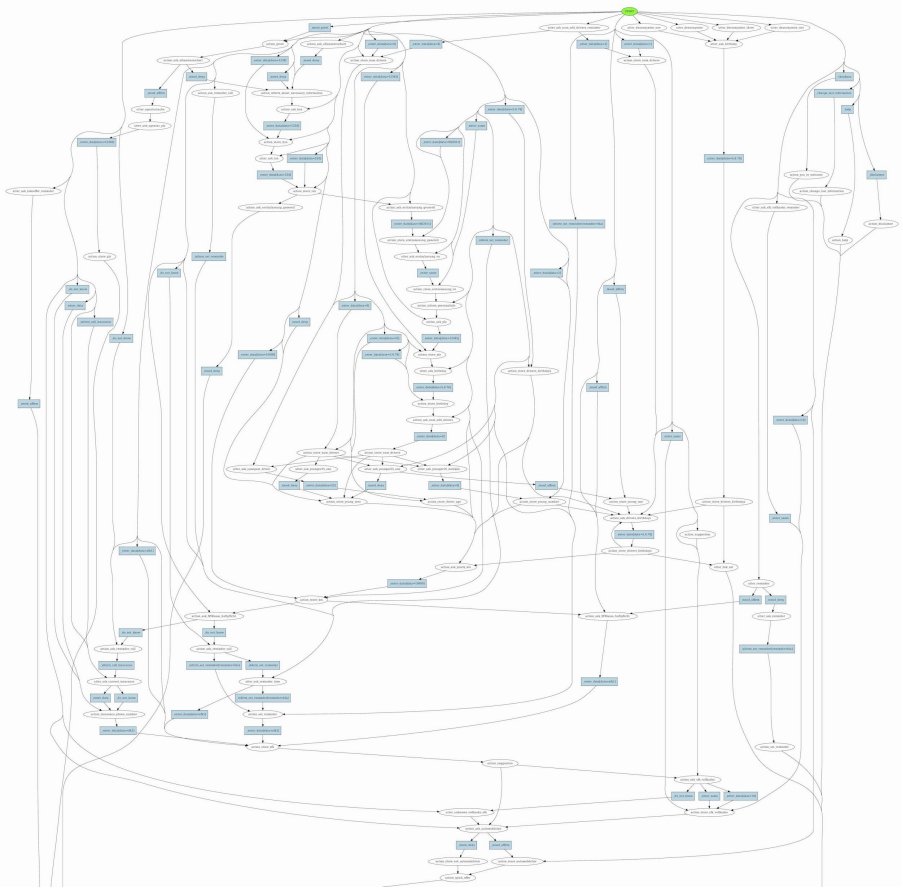
Why Dialogue Handling with Rasa Core?

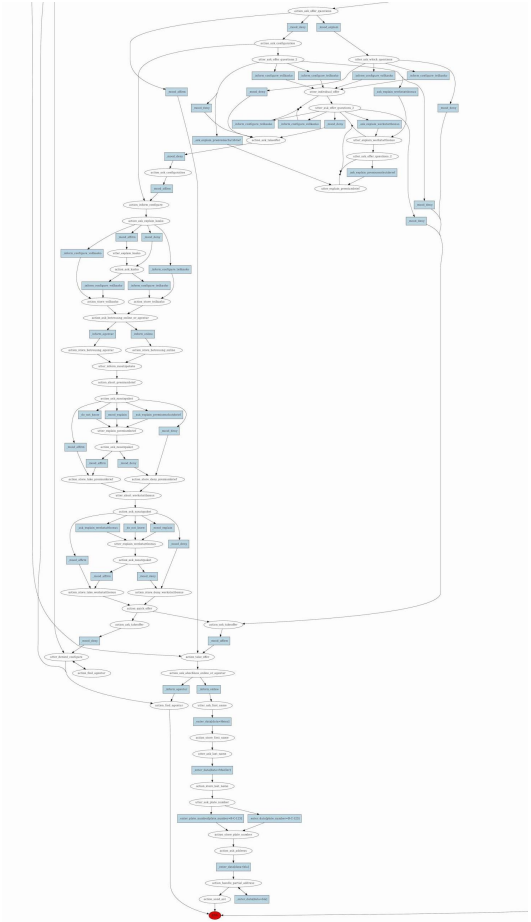
- No more state machines!
- Reinforcement Learning: too much data, reward functions...
- Need a simple solution for everyone

Why Machine Learning?



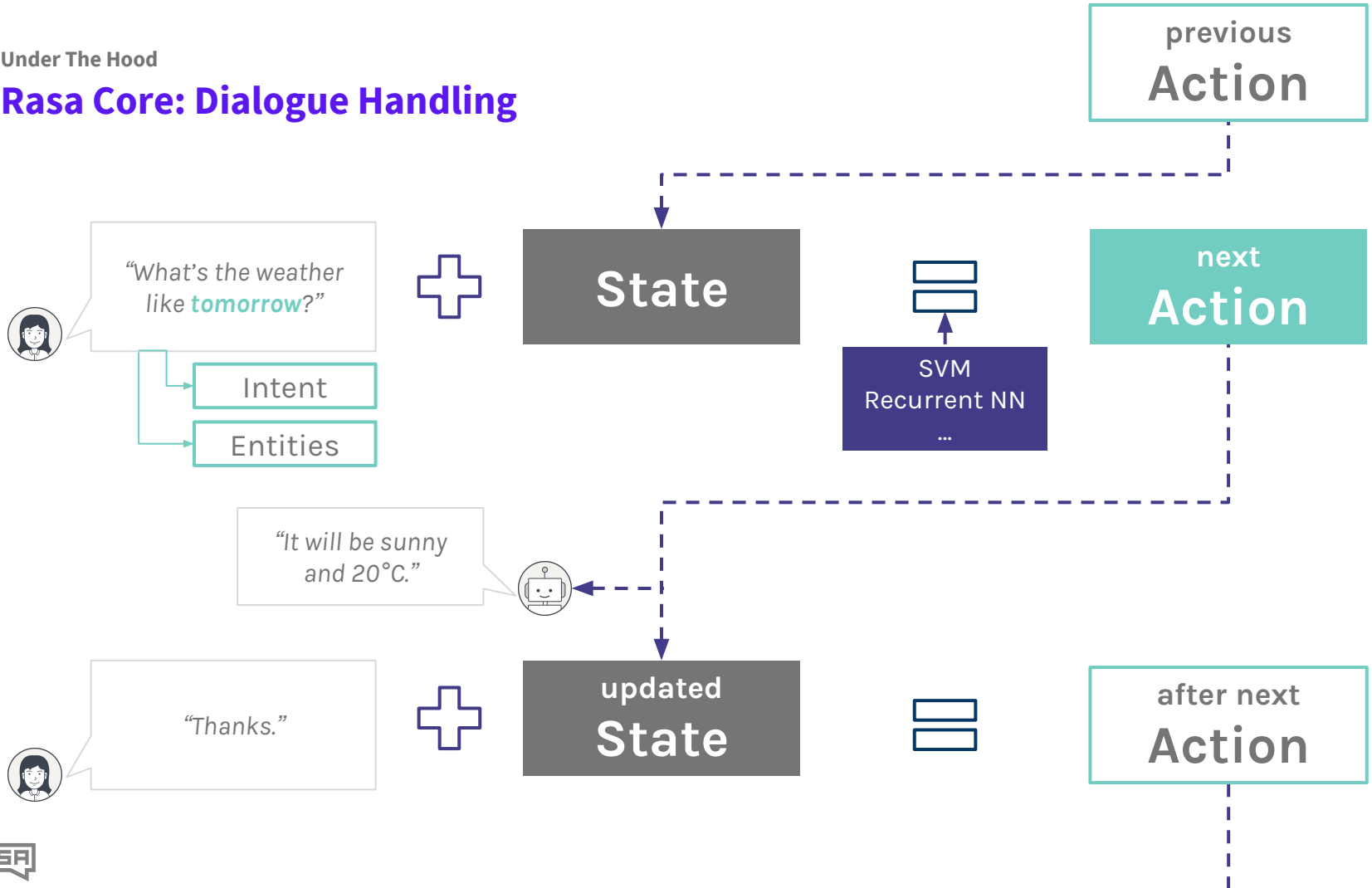
State Machines are infeasible





Under The Hood

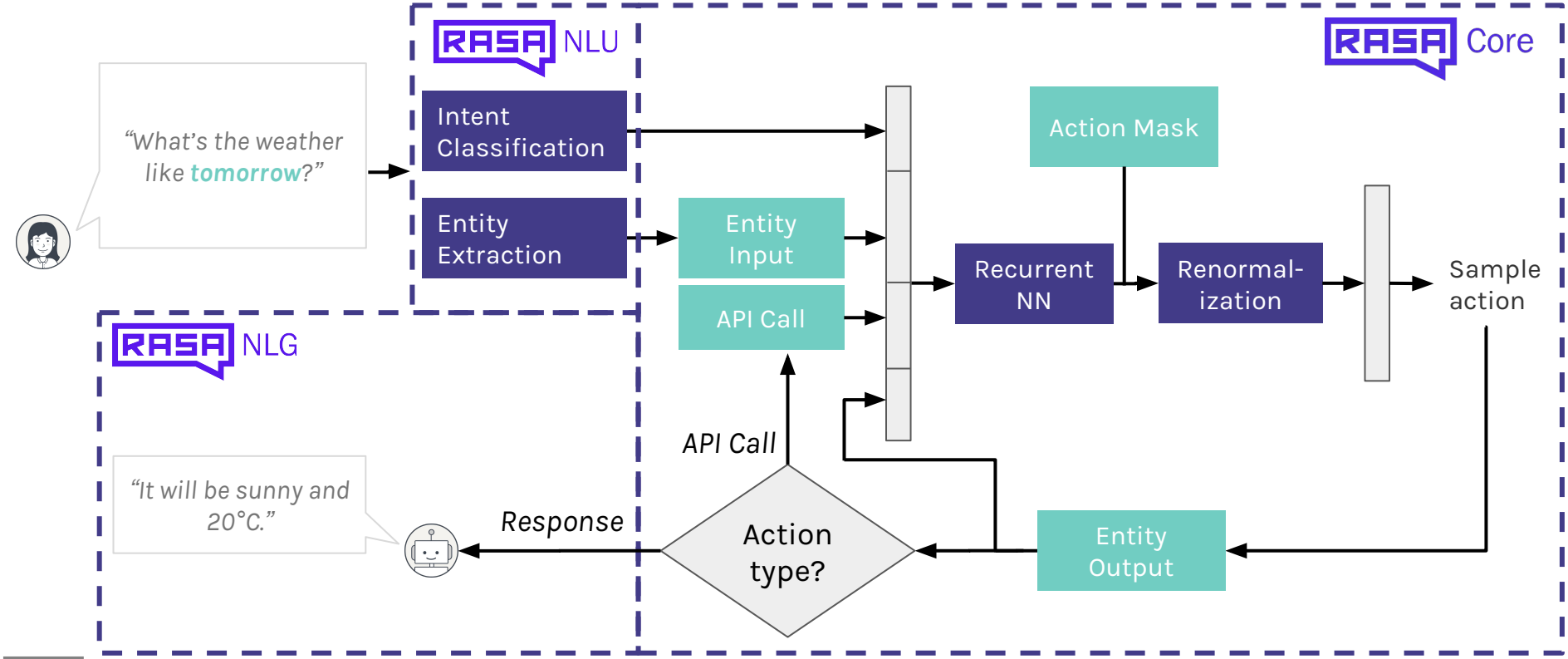
Rasa Core: Dialogue Handling



Under The Hood

Rasa Core: Dialogue Handling

Similar to LSTM-dialogue prediction paper: https://arxiv.org/abs/1606.01269



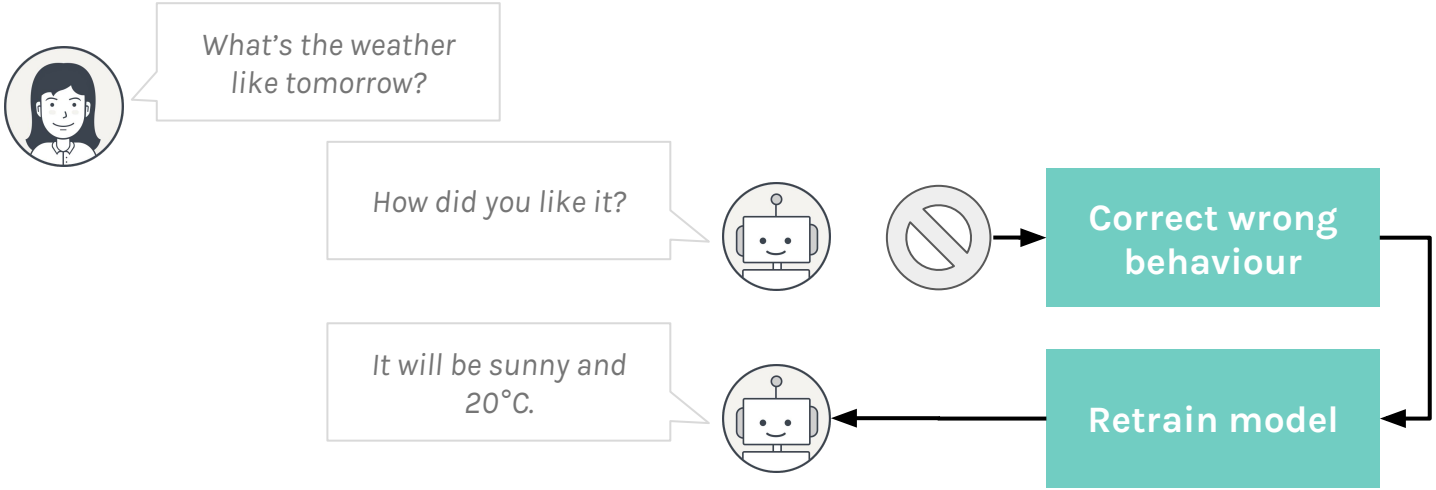
Let’s Code

Dialogue Engine

RASA

Rasa Core: Dialogue Training

Issue: How to get started? → Online Learning



Let’s Code

Interactive Learning

RASA

Research

Training NLU models without initial word vectors

Goal: Learn an **embedding** for the intent labels based on the user messages

- Learns joined embeddings for intents & words at the same time
- Allows multi-intent labels
- Knows about similarity between intent labels
- Based on Starspace Paper

https://medium.com/rasa-blog/supervised-word-vectors-from-scratch-in-rasa-nlu-6daf794efcd8

Training NLU models without initial word vectors

Goal: Learn an **embedding** for the intent labels based on the user messages

Multi-Intent:

Text	Intent
Hey how are you? i don't really care	`greet+dontcare`
ok something else then? thanks a bunch	`deny+thankyou`
cool! Who is the mayor or New York City?	`state_happy+random`

Evaluation:

Pipeline	train F1-score	test F1-score
spacy (small)	0.684 (0.020)	0.325 (0.018)
tensorflow_embedding	0.984 (0.001)	**0.898** (0.017)

Generalisation *across* dialogue tasks

Why do we need this complex architecture? For generalisation between domains!

```
## hotel explain 1.3
* request_hotel
    - utter_ask_details
* inform{"location": "paris"}
    - utter_ask_people
* inform{"people": "4"}
    - utter_ask_price
* explain
    - utter_explain_price_hotel
    - utter_ask_price
```

```
## restaurant explain 1.3
* request_restaurant
    - utter_ask_details
* inform{"location": "paris"}
    - utter_ask_people
* inform{"people": "4"}
    - utter_ask_price
* explain
    - utter_explain_price_restaurant
    - utter_ask_price
```

Extensions

Pre-Trained Entity Models

Rule Based Models (e.g. Duckling):

- Extract entities with restricted formats using e.g. patterns
- Allow for normalization (e.g. “tomorrow” → 26.05.18)

Reusing Trained ML models (e.g. spaCy builtins):

- Trained on large corpora on common entities (persons, cities)
- Usually restricted to the language they are trained on

Form Filling

- Make some form logic deterministic
- Request all fields and then call API
- Simplifies action space

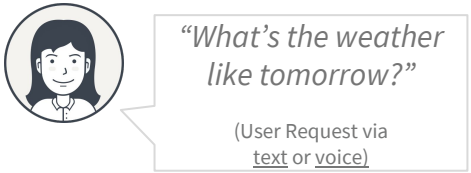
```
* request_restaurant
    - action_restaurant_form
    - slot{"requested_slot": "people"}
* inform{"number": 3}
    - action_restaurant_form
    - slot{"people": 3}
    - slot{"requested_slot": "time"}
* inform{"time": "8pm"}
   - action_restaurant_form
```

```python
class ActionSearchRestaurants(FormAction):

    RANDOMIZE = False

    @staticmethod
    def required_fields():
        return [
            EntityFormField("cuisine", "cuisine"),
            EntityFormField("number", "people"),
            BooleanFormField("vegetarian", "affirm", "deny")
        ]

    def name(self):
        return 'action_search_restaurants'

    def submit(self, dispatcher, tracker, domain):
        results = RestaurantAPI().search(
            tracker.get_slot("cuisine"),
            tracker.get_slot("people"),
            tracker.get_slot("vegetarian"))
        return [SlotSet("search_results", results)]
```

Final Thoughts

Final Thoughts

Closing The Loop



New Training Data

Correct Training Data

Retrain ML Models

Relabel Collected Data

Use Improved Model

Open challenges

For those that are curious:

- Handling OOV words
- Multi language entity recognition
- Combination of dialogue models

We're constantly working on improving our models!

Current Research

Good reads for a rainy day:

- Last Words: Computational Linguistics and Deep Learning (blog)
  https://goo.gl/lGSRuj
- Starspace Embeddings (paper)
  https://arxiv.org/abs/1709.03856
- End-to-End dialogue system using RNN (paper)
  https://arxiv.org/pdf/1604.04562.pdf
- MemN2N in python (github)
  https://github.com/vinhkhuc/MemN2N-babi-python
- Sentence Embeddings (blog)
  https://medium.com/huggingface/universal-word-sentence-embeddings-ce48ddc8fc3a

Summary

4 take home thoughts:

- Techniques to handle small data sets are key to get started with conversational AI
- Deep ML techniques help advance state of the art NLU and conversational AI
- Combine ML with traditional Programming and Rules where appropriate
- Abandon flow charts

Thanks!

Tom Bocklisch
Head of Engineering
tom@rasa.com