From zero to hero: building contextual AI assistants with machine learning and Rasa OSS tools

Justina Petraityte, Head of Developer Relations @ Rasa

What are we focusing on during this workshop

Goal:

Build a ML-powered AI assistant

Roadmap:

1. Intro to conversational AI and Rasa
2. Natural Language Understanding
    i. Introduction and theory
    ii. Coding
3. Dialogue Handling
    i. Introduction and theory
    ii. Coding
4. Enabling your assistant to learn continuously

Setup

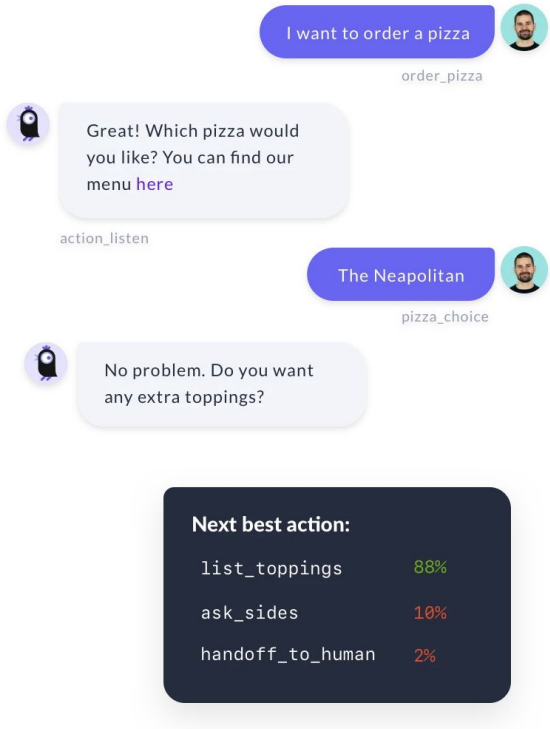
1. Environment with Python 3 installed
2. Repository: https://github.com/JustinaPetr/botsbrasil
3. Install:
   a. Rasa and Rasa X (instructions in a README.md of the project repository)
   b. Ngrok (optional)

Intro to Conversational AI and Rasa

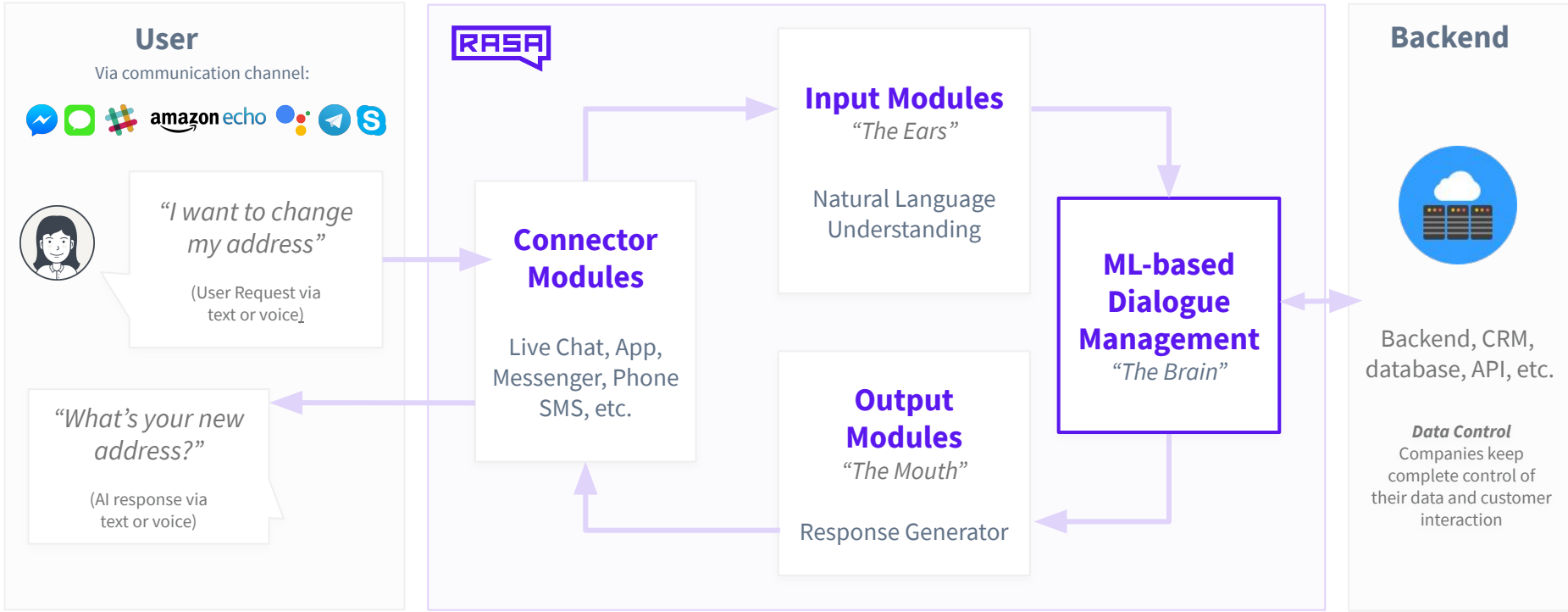
Rasa: Open source tools for developers to build text and voice assistants in-house

Rasa is a set of open source machine learning tools for developers for conversational AI:

- **NLU**: a an open-source natural language processing tool for intent classification and entity extraction
- **Core**: framework for machine learning-based, contextual decision making

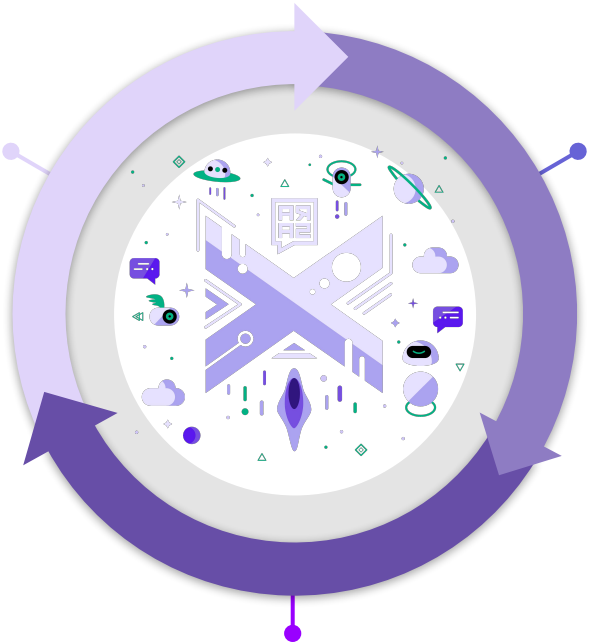


Rasa's technology can understand natural language and decide about the next best action based on the context of the conversation using Machine Learning



Improving your assistant using real conversational data with Rasa X

Deploy an assistant in production



Collect real conversations between the users and your assistant.

Use collected conversational data to improve your assistant

Developing AI assistants with Rasa

Rasa NLU: Natural Language Understanding

The goal of Rasa NLU is to extract the structured data from unstructured user inputs

Data:

intent: order_pizza
- I would like to order a large pizza
- Want some pizza please

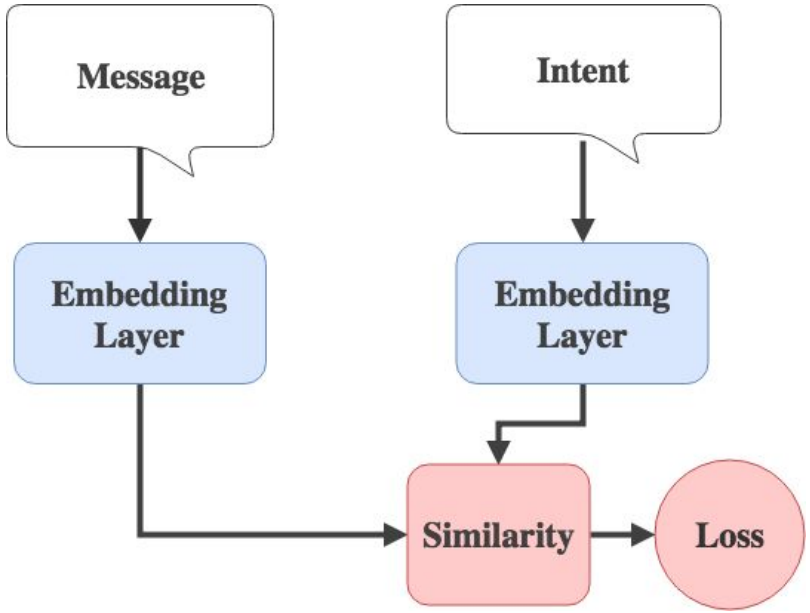
intent: greet
- Hello
- hi

Supervised Word Vectors from scratch

References:

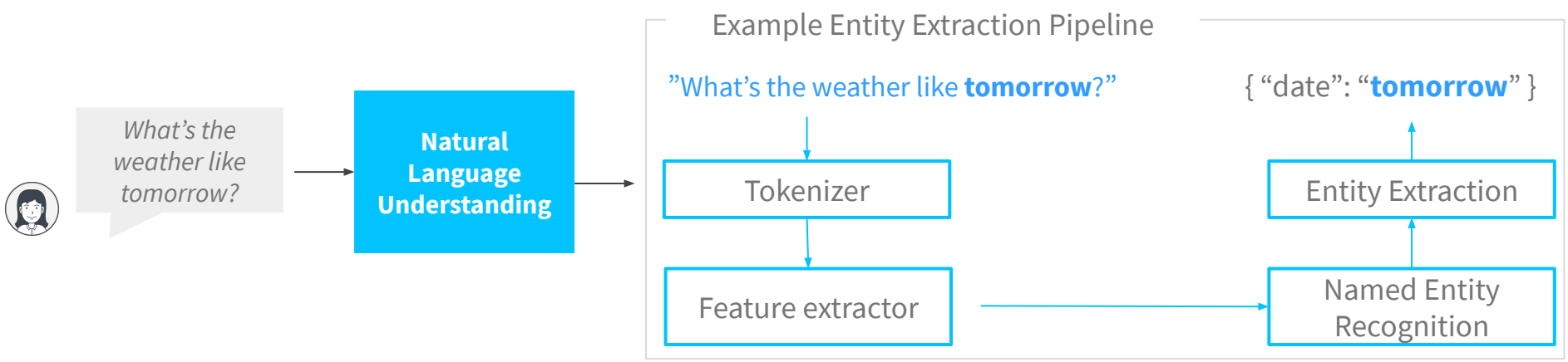
WSABIE (Weston, Bengio, Usunier)

StarSpace (Wu et al)

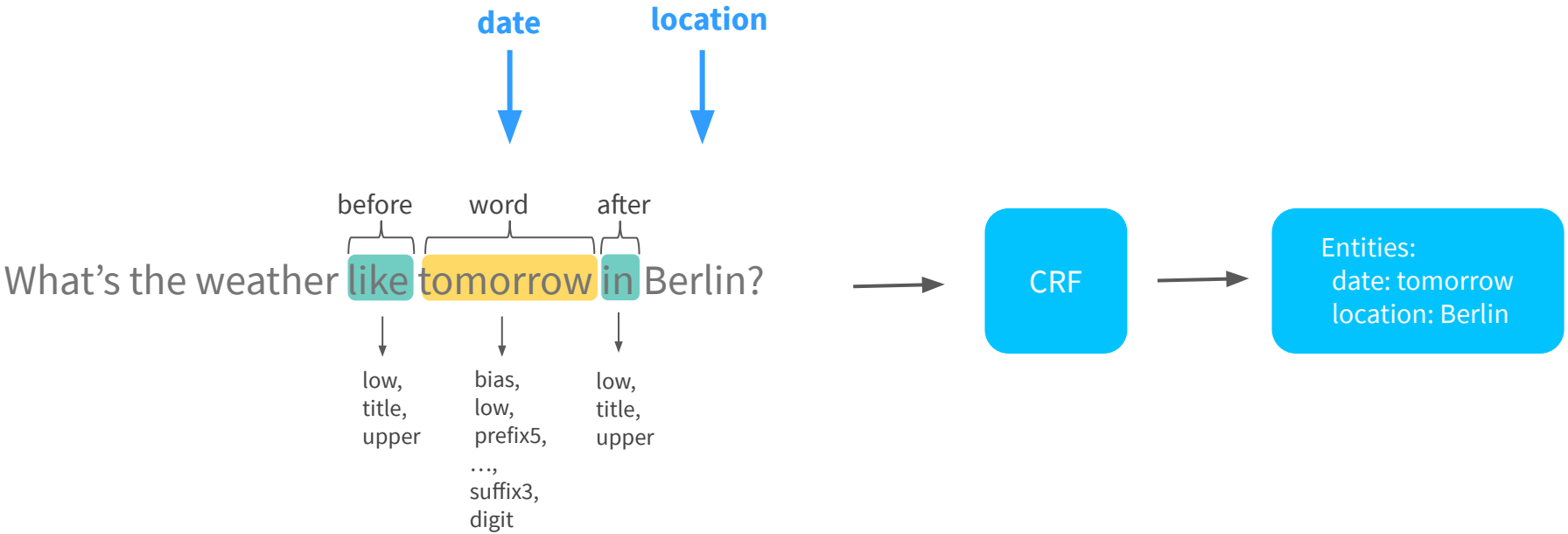


$$\sum_{(a,b)\in \mathbf{E}^+} \sum_{b^-\in \mathbf{E}^-} L^{batch}\left(\text{sim}(a,b), \text{sim}(a,b_1^-), \ldots, \text{sim}(a,b_k^-)\right)$$

Rasa NLU: Entity Extraction



Rasa NLU: Entity Extraction



Challenges

Handling synonyms

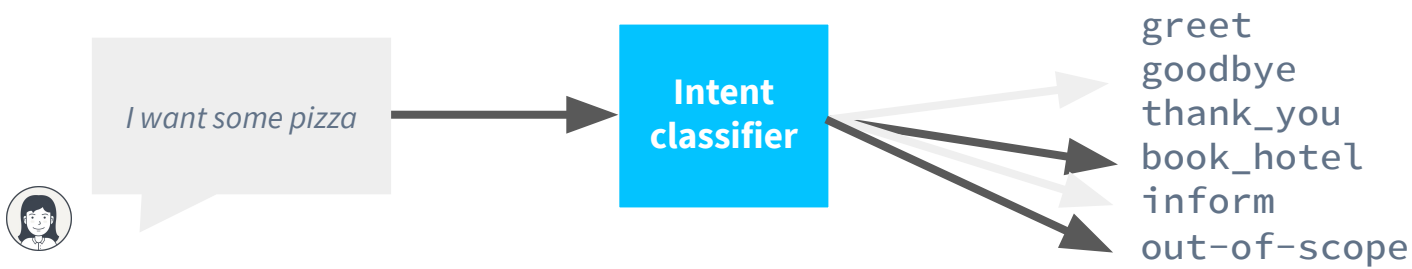
I moved to New York City.

I moved to NYC.


```
[{
  "text": "I moved to New York City",
  "intent": "inform_relocation",
  "entities": [{"value": "nyc",
                "start": 11,
                "end": 24,
                "entity": "city",
               }]
},
{
  "text": "I got a new flat in NYC.",
  "intent": "inform_relocation",
  "entities": [{"value": "nyc",
                "start": 20,
                "end": 23,
                "entity": "city",
               }]
}]
```

Identifying out-of-scope inputs

The best way to handle out-of-scope inputs is to actually teach your bot to identify them.



Understanding multi-intents

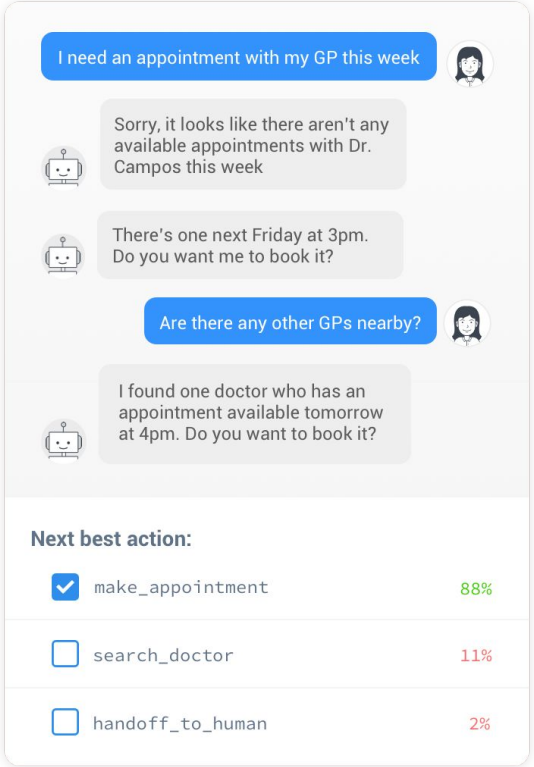
A user input can have more than one intention. Enabling the assistant to understand them leads to more natural conversations.

Message	Intent
Sounds great! Can you also tell me what is the price?	affirm+ask_price
What about tomorrow? I feel too tired today.	inform+chitchat
Yes, book it. Also, please book me a taxi.	affirm+book_taxi

Dialogue Management

Why Dialogue Handling with Core?

- Learn from real conversational data
- Close the feedback loop using the real-time user feedback
- Prototype quickly and start with very little training data
- Leverage the power of ML to build assistants that scale in production



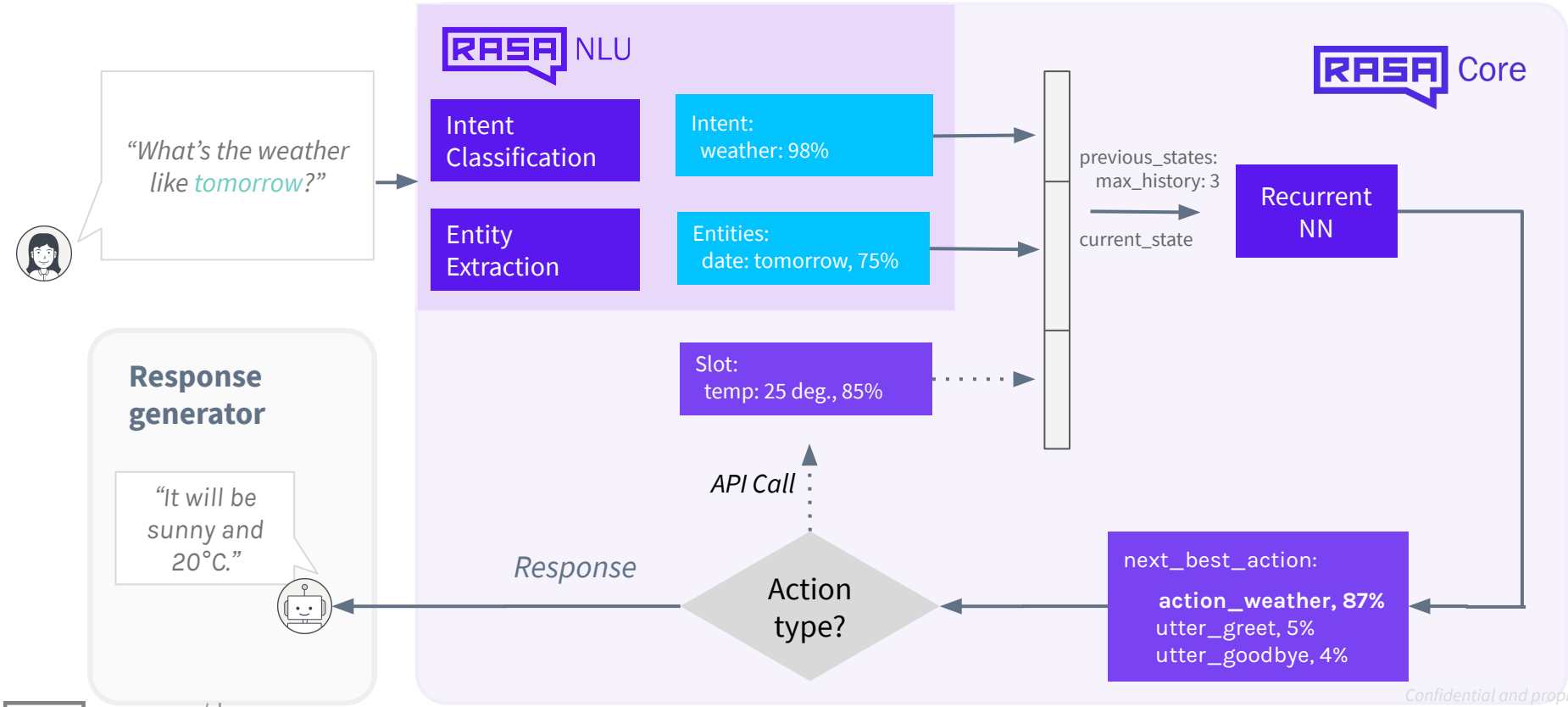
Core learns to Converse from real conversational data

U: Hello
B: Hello. I am Sara and I would like to help you get started with Rasa. What is your name?
U: I am Juste
B: And where are you from?
U: From Berlin
B: Have you used Rasa before?
U: Yes.
B: Then, you should check out our latest blog.
U: Thanks.

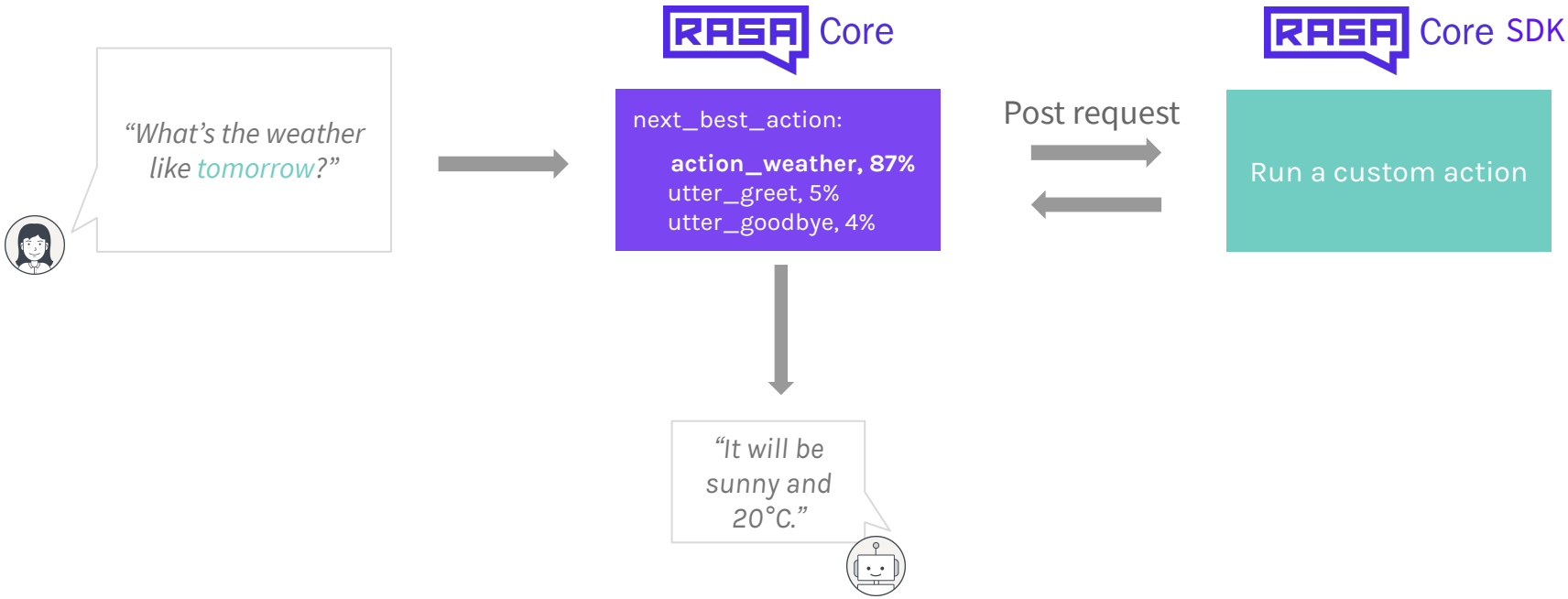
*greet
- utter_greet
* name
- utter_ask_location
* location
- utter_used_rasa
* affirm
- utter_send_blog
* thanks

Rasa Core: Dialogue Handling

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



Let's zoom in on a action server

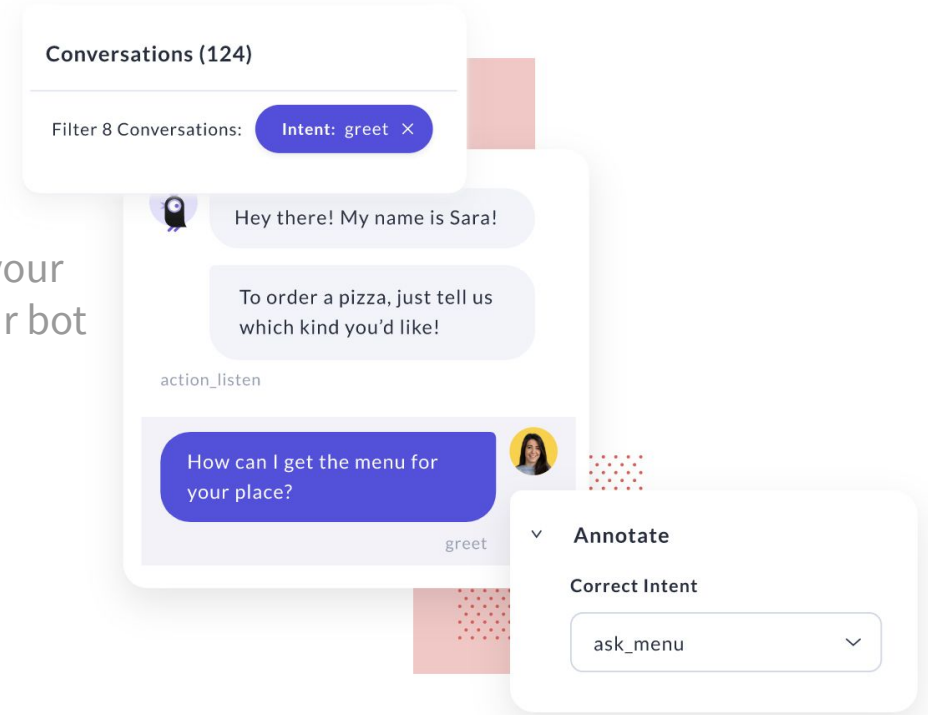


Enable your assistant to learn from real conversational data

Under The Hood

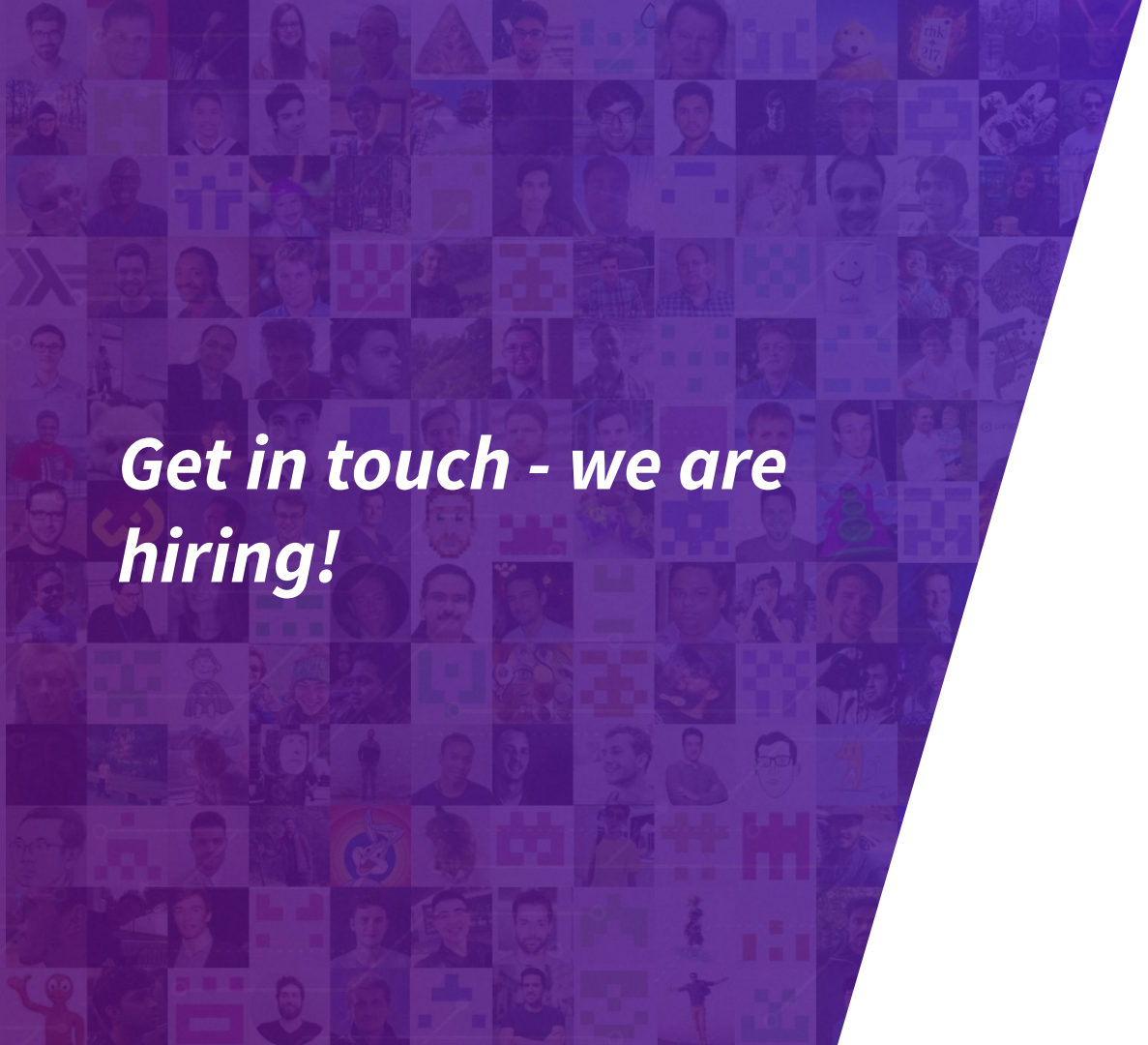
Use real conversations to improve your assistant

Collect conversations between real users and your assistant, correct and use them to improve your bot over time.



Key takeaways

- Allow your assistant to learn from real conversations early on
- Start small and make sure your assistant nails the happy path first
- Leverage the freedom of customization the open source tools give you
- When needed - use business logic to build your best conversational AI



Get in touch - we are hiring!

Justina Petraityte

Head of Developer Relations

juste@rasa.com

@juste_petr