Building contextual AI assistants using machine learning and open source tools

Ty Dunn, Product Manager @ Rasa
Rasa is building the standard infrastructure for conversational AI

Level 1: Notifications
Level 2: “Chatbots” - FAQs
Level 3: Contextual Assistants
Level 4: Personalized Assistants
Level 5: Autonomous Organization
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**: an open-source natural language processing tool for intent classification and entity extraction
- **Core**: framework for machine learning-based, contextual decision making
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
WHY RASA

Why do developers love Rasa today?

Customizable

Tune models and get higher accuracy on your data set. Integrate into your existing infrastructure.

Open source and free

No risk of vendor lock-in. No “AI fairy dust” - can see all ML components.

Empowers developers

Get started with very little ML knowledge, look under the hood and learn more about ML.
Join the Rasa open source community!

Open Source + Community + Applied Research
Understanding Rasa
Rasa’s technology can understand natural language and decide about the next best action based on the context of the conversation using Machine Learning.
Under the Hood

**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
Under the Hood

**Supervised Word Vectors from scratch**

References:

**WSABIE** (Weston, Bengio, Usunier)

**StarSpace** (Wu et al)

\[
\sum_{(a,b) \in E^+, b^- \in E^-} L^{batch}(\text{sim}(a, b), \text{sim}(a, b_1^-), \ldots, \text{sim}(a, b_k^-))
\]
# Understanding multi-intents

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

<table>
<thead>
<tr>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sounds great! Can you also tell me what is the price?</td>
</tr>
<tr>
<td>What about tomorrow? I feel too tired today.</td>
</tr>
<tr>
<td>Yes, book it. Also, please book me a taxi.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intent</th>
</tr>
</thead>
<tbody>
<tr>
<td>affirm+ask_price</td>
</tr>
<tr>
<td>inform+chitchat</td>
</tr>
<tr>
<td>affirm+book_taxi</td>
</tr>
</tbody>
</table>
Rasa NLU: Entity Extraction

Under the Hood

Example Entity Extraction Pipeline

"What’s the weather like tomorrow?"

Tokenizer

Feature extractor

Entity Extraction

Named Entity Recognition

{ “date”: “tomorrow” }
What's the weather like tomorrow in Berlin?
Context is king - NLU is not enough to handle that

## New story
* greet
  - utter_greet_user
  - utter_ask_name

Next best action:

<table>
<thead>
<tr>
<th>Action</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>utter_ask_name</td>
<td>89%</td>
</tr>
<tr>
<td>utter_greet_user</td>
<td>11%</td>
</tr>
</tbody>
</table>
Why Dialogue Handling with Core?

Under The Hood

- Learn from real conversational data instead of enforcing rules
- Leverage the power of ML to build assistants that scale in production
- Hold back-and-forth conversations that remember context
- Add business logic when needed
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
“What’s the weather like tomorrow?”

Intent Classification: weather: 98%

Entity Extraction: date: tomorrow, 75%

Slot: temp: 25 deg., 85%

Response generator:

“It will be sunny and 20°C.”

next_best_action:
- action_weather, 87%
- utter_greet, 5%
- utter_goodbye, 4%

Enable your assistant to learn from real conversational data
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.
Let’s see how it all works in practice!

It’s demo time!
Key takeaways and tips
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
Compressing BERT to make it fast (to train and run)

Transfer learning of reusable dialogue patterns

Mixing retrieval-based and task-oriented dialogue
Become a contributor to Rasa!

Rasa Contributor Program

The Rasa Contributor program consists of three stages, each based on different levels of involvement and each with a bunch of different perks:

**Contributor**

**WHO ARE THEY?**
Contributors are community members who help us improve the Rasa software by contributing to Rasa codebase or developer resources.

**Hero**

**WHO ARE THEY?**
Rasa Heroes are experienced Rasa contributors who have made multiple high-quality contributions and champion Rasa within other communities.

**Superhero**

**WHO ARE THEY?**
Rasa Superheroes are Rasa community leaders who grow Rasa community in their local area and bring people interested in conversational AI together.
Rate today’s session
Get in touch - we are hiring!

Ty Dunn
Product Manager
t.dunn@rasa.com
@tylerjdunn