Making chatbots using rasa

Lianghui Yi

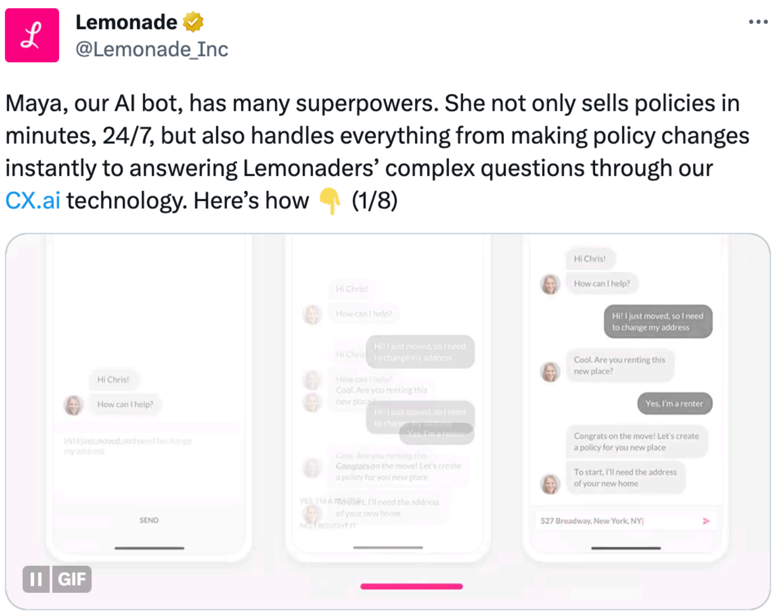
What are chatbots?

- Siri
- Alexa
- ChatGpt

Hi, everyone! Today I gonna introduce chatbots. Since they are so popular, I guess most of you have used them before. For example, Siri, Alexa, and ChatGpt. So chatbots are just applications that can talk to.

What can chatbots do?

- Customer Service (FAQ)
- Insurance Sales (Lenmonade, conversational)
- ...



Lemonade
@Lemonade_Inc

Maya, our AI bot, has many superpowers. She not only sells policies in minutes, 24/7, but also handles everything from making policy changes instantly to answering Lemonaders' complex questions through our CX.ai technology. Here's how 👇 (1/8)

Hi Chris!

How can I help?

Hi! I just moved, so I need to change my address

Cool. Are you renting this new place?

Yes, I'm a renter

Congrats on the move! Let's create a policy for you new place

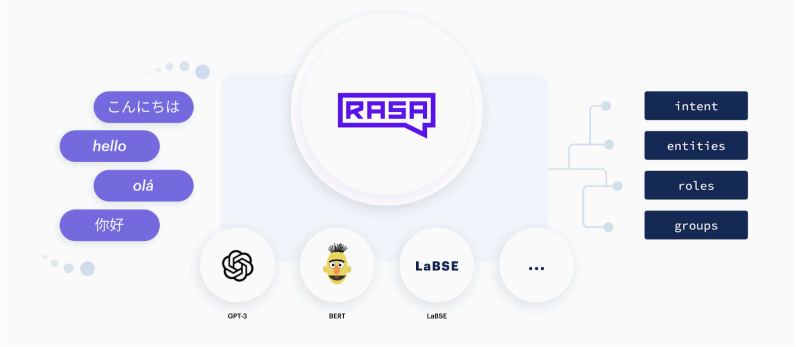
To start, I'll need the address of your new home

527 Broadway, New York, NY

So, what can chatbots do? Actually, they are quitely popular under the hood. For example, When you are asking some questions about a product on a website, you are probably talking to a chatbot. For example, Lemonade is an insurance company that uses chatbots to sell insurance.

What is Rasa

With over 25 million downloads, Rasa Open Source is the most popular open source framework for building chat and voice-based AI assistants.



So, what is Rasa? You can use this framework to build your own chatbot. It's open source and free. By using this tool, you can even create chatbot without writing any code.

How to Install Rasa

```
pip3 install -U pip
pip3 install rasa
```

Now, let's install Rasa. You can easly install it by using pip. Be careful, the python version should not be too new.

Notes: Python 3.10> is not supported, you should downgrade to early version or using conda to switch.

Play with cli

- rasa init (create a new project)
- rasa train (train a model)
- rasa shell (talk to your assistant)
- rasa interactive (teach your assistant with interactive learning)

After installation, let's play with the cli. Here are a list of commands you can use. First, you can use rasa init to create a new project. Then, you can use rasa train to train a model. After that, you can use rasa shell to talk to your assistant.

https://rasa.com/docs/rasa/command-line-interface

Train your model

- Intents
- Responses
- Rules
- Stories

Now, let's talk about how to train your model.

Intents

What Does the User Say.

```yaml
version: "3.1"
nlu:
- intent: greet
  examples: |
    - hey
    - hello
    - hi
    - hello there
    - good morning
    - good evening
```

First, Intents are what the user says. For example, when the user says “hey”, “hello”, “hi”, the intent is greet.

Response

How the bot replies to the query.

```yaml
responses:
  utter_greet:
  - text: "Hey! How are you?"

  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!"
```

Responses are how the bot replies to the query. You can use text, image, and even emoji in your response.

Rules

Basic rules that match intents and reponses

```
version: "3.1"

rules:

- rule: Say goodbye anytime the user says goodbye
  steps:
  - intent: goodbye
  - action: utter_goodbye

- rule: Say at meeting when asked what doing by girlfriend
  steps:
  - intent: gf_doing
  - action: utter_meeting
```

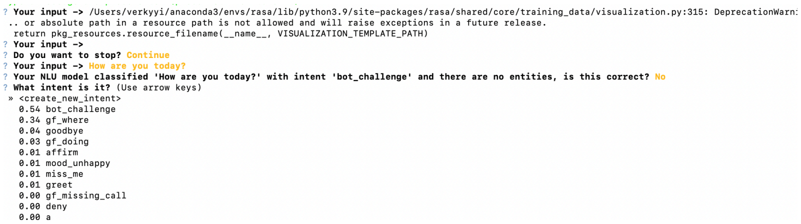
Then, let's talk about rules. Rules are basic rules that match intents and responses. For example, when the user says A, the bot will reply B. And A or B Could be whatever intents and responses you defined in advance.

Story

Stories are common conversation flows. Beyond rules, this can give more context information to the bot. So they can be more clever.

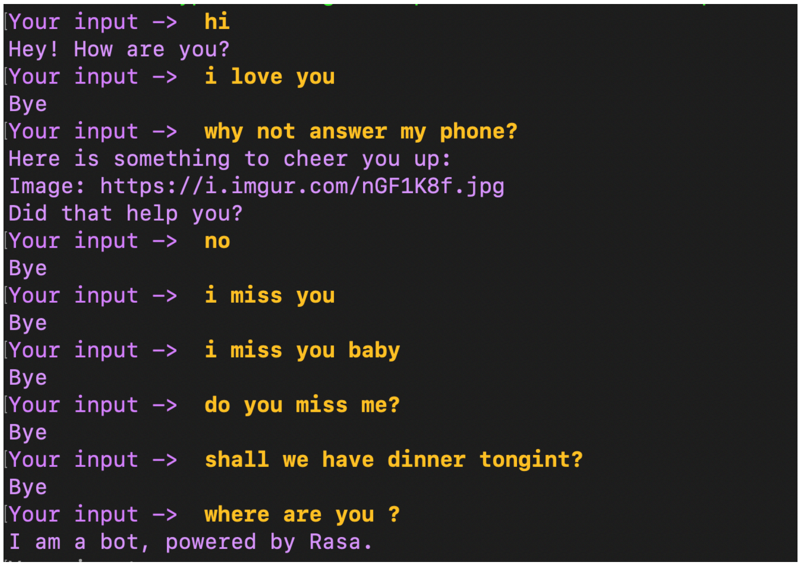
Common conversation flows

```
version: "3.1"
stories:
- story: happy path
  steps:
  - intent: greet
  - action: utter_greet
  - intent: mood_great
  - action: utter_happy
```

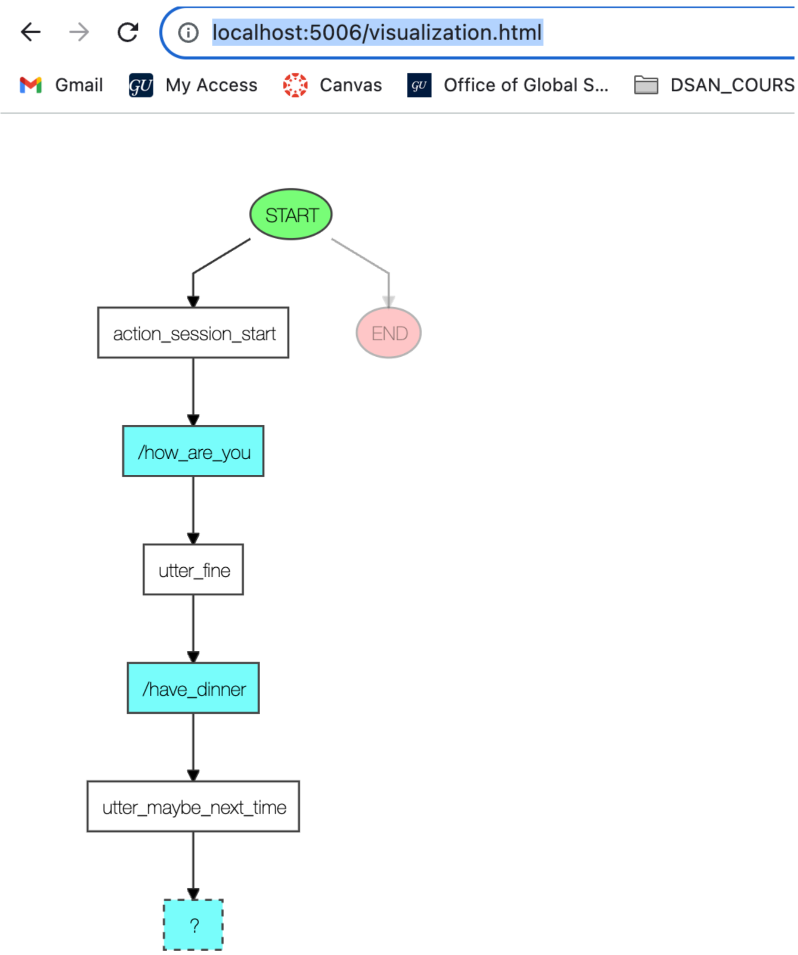


```
? Your input -> /Users/verkyyi/anaconda3/envs/rasa/lib/python3.9/site-packages/rasa/shared/core/training_data/visualization.py:315: DeprecationWarn:
 .. or absolute path in a resource path is not allowed and will raise exceptions in a future release.
  return pkg_resources.resource_filename(__name__, VISUALIZATION_TEMPLATE_PATH)
? Your input ->
? Do you want to stop? Continue
? Your input -> How are you today?
? Your NLU model classified 'How are you today?' with intent 'bot_challenge' and there are no entities, is this correct? No
? What intent is it? (Use arrow keys)
 » <create_new_intent>
   0.54 bot_challenge
   0.34 gf_where
   0.04 goodbye
   0.03 gf_doing
   0.01 affirm
   0.01 mood_unhappy
   0.01 miss_me
   0.01 greet
   0.00 gf_missing_call
   0.00 deny
   0.00 a
```

Debug using real time conversation graph



```
Your input ->  hi
Hey! How are you?
Your input ->  i love you
Bye
Your input ->  why not answer my phone?
Here is something to cheer you up:
Image: https://i.imgur.com/nGF1K8f.jpg
Did that help you?
Your input ->  no
Bye
Your input ->  i miss you
Bye
Your input ->  i miss you baby
Bye
Your input ->  do you miss me?
Bye
Your input ->  shall we have dinner tongint?
Bye
Your input ->  where are you ?
I am a bot, powered by Rasa.
```



When you already trained your model, before publish to the public, you might need to debug your bot. Rasa provides a real time conversation graph to help you debug. As you can see on the pictures, it shows what the bot is thinking while talking with the bot.

Demo time

To demo the easyness of Rasa, I build a simple chatbot in a imagined scenario. Suppose I have a girfriend, and she always text me or call while I am busy at meeting. So I build a chatbot to answer her questions.

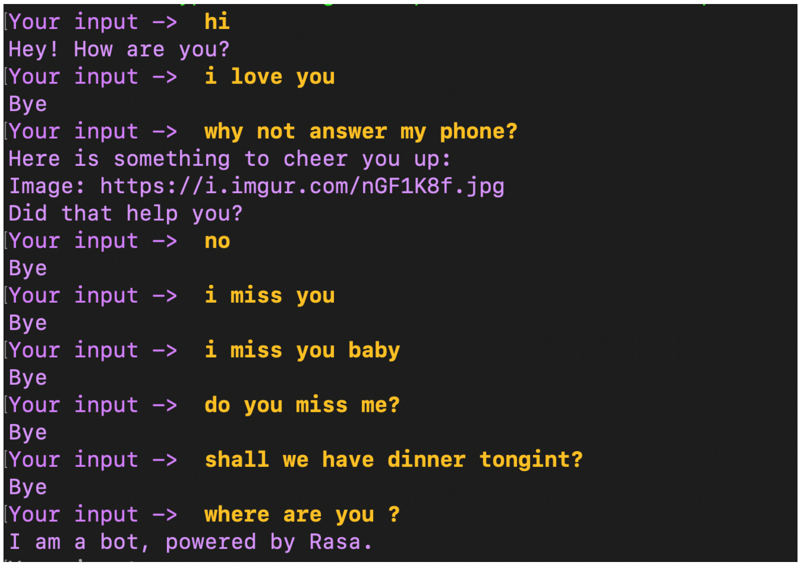
Frist we can see these are the training files I wrote. which are just intents responses rules and stories.

Since I already trained the model in advance, so now i will use rasa shell to test the result.

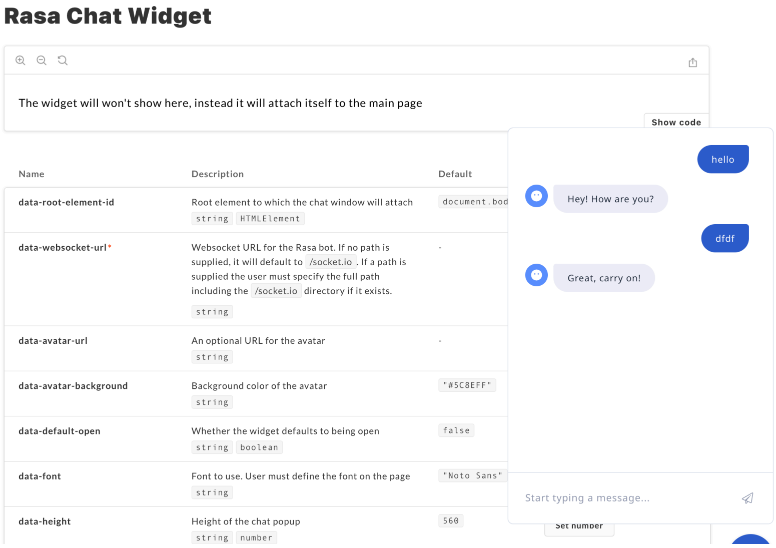
- **why don't answer the call?** at meeting.
- **would you have dinner with me tonight?** maybe next time, baby
- **okay, i miss you.** miss you, too. will let you know when I was free.
- **are you a robot?** of course not. how dare I? ... That's it.

A simple chatbot that can help you answer questions coming from your girl friend.

```
# Using Shell
rasa shell
# Using Web widget
rasa run --enable-api --cors "*" --debug
```



```
Your input ->  hi
Hey! How are you?
Your input ->  i love you
Bye
Your input ->  why not answer my phone?
Here is something to cheer you up:
Image: https://i.imgur.com/nGF1K8f.jpg
Did that help you?
Your input ->  no
Bye
Your input ->  i miss you
Bye
Your input ->  i miss you baby
Bye
Your input ->  do you miss me?
Bye
Your input ->  shall we have dinner tongint?
Bye
Your input ->  where are you ?
I am a bot, powered by Rasa.
```



Rasa Chat Widget

The widget will won't show here, instead it will attach itself to the main page

Name	Description	Default
data-root-element-id	Root element to which the chat window will attach `string` `HTMLElement`	`document.bod`
data-websocket-url*	Websocket URL for the Rasa bot. If no path is supplied, it will default to `/socket.io`. If a path is supplied the user must specify the full path including the `/socket.io` directory if it exists. `string`	-
data-avatar-url	An optional URL for the avatar `string`	-
data-avatar-background	Background color of the avatar `string`	`"#5C8EFF"`
data-default-open	Whether the widget defaults to being open `string` `boolean`	`false`
data-font	Font to use. User must define the font on the page `string`	`"Noto Sans"`
data-height	Height of the chat popup `string` `number`	`560`

hello

Hey! How are you?

dfdf

Great, carry on!

Summary

So, here are the summary.

- Rasa is a powerful open source framework for building simple chatbots.
- It’s easy to use but needs a lot of training data.
- You can publish to public channels like slack discord telegram, etc.

By the way, if you really have a girlfriend, do not try to use this dummy chatbot to fool her.

- Rasa is a powerful open source framework for building chatbots.
- Good robots needs a lot of training data.
- You can publish to public channels like slack discord telegram, etc.