

Career Bot- Using Rasa NLU

-Shreyas Torgalmath(<https://spotle.ai/ShreyasTorgalmath>)

Contents:

- Definition and meaning of some file
- What is Rasa
- Installation of Rasa
- How to train chatbot using Rasa-NLU
- Uses of the bot
- Conclusion

Definition and meaning of some file

Intent: An **intent** is an abstract description of an operation to be performed.

Eg. Hi,Hello,hey are greetings. So we can call Greeting as Intent.

Entity: Entity is used to get useful information from users.

Actions:Actions gives response to user input.

Pipeline: The data will flow and intent classification & entity extraction can be done for the bot in the Pipeline

Templates:It's the message that bot response to the user depending on the user input.

Eg: user:Hi

Bot: hello, How can I help you?

“Hello,How can I help you” is the template

.yaml:YAML is a human-readable data-serialization language. It is commonly used for configuration files, but could be used in many applications where data is being stored or transmitted

.md:An MD file is a text file created using one of several possible dialects of the Markdown language

.py:Python file

Files used

nlu.md

nlu_model.py

domain.yml

stories.md
policy.yml
endpoints.yml
actions.py

nlu.md

It contains the training data in terms of user inputs along with intent and entities .As the number of examples increases the better the chatbot work

nlu_model.py

It's the python file which contains the code for training the chatbot. This will train the NLU model and create some data which is useful data for training the bot.

domain.yml

The domain contains of five key parts

Intent

entities

Slots

Actions

Templates

stories.md

In domain.yml we have defined Intent and template for it.In stories.md we will decide for what intent,which template should reply.

policy.yml

How every company,school has set of policy, Same way Rasa NLU has set of Policy which to decide which action to take at every step in the conversation.

endpoints.yml

To run our bot we need server, this will specify which server we need to use it.

actions.py

After creating everything we need to run the bot. We need to create custom actions to run the bot. We use rasa_core_sdk for the actions.

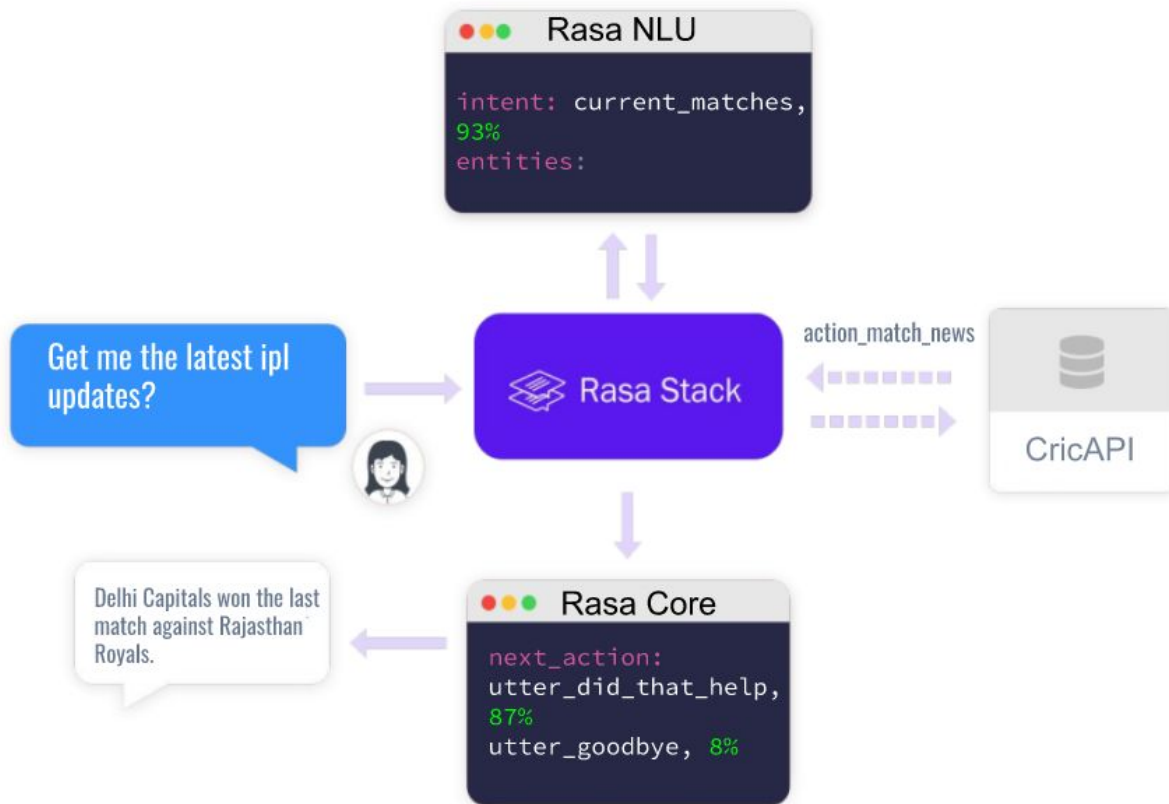
What is Rasa?

Rasa is an open source Conversational AI framework.Being an opensource framework, it is easy to customize.

It has 2 main Components :

Rasa NLU (Natural Language Understanding) :It is a tool for understanding what is being said in short pieces of text.It extracts the input from user and classifies the intent

Rasa Core:It is used to guide the flow of conversation



Pic Credits : analyticsvidhya

In this picture we can understand that

When user inputs the data into the Rasa Stack
Rasa NLU classifies the intent and entities
Rasa core decides the action for it.

Installation of pre required softwares

Before installing Rasa, we need to install Microsoft C++

(<https://visualstudio.microsoft.com/downloads/>) . In this goto tools for visual studio 2019.

You can find Build tools for Visual studio.

After installing this, Open Visual Studio than Install C++ in it. It will be about 1.08GB.
After installing this, We can go for further
We need to go Create virtual environment to run our Bot.

Virtual Environment

Lets create the virtual Environment

I recommend using anaconda prompt to perform this, You can use Command prompt, But I like anaconda prompt, so I am working in anaconda prompt

```
conda create --name botenv python=3.6
```

I have created python virtual environment, I have kept its name as botenv which means bot-environment. You can create with any name, Better use standard name.

To activate bot we need to use

```
conda activate botenv
```

To deactivate we need to use

```
conda deactivate
```

Installing of Rasa NLU

It's time to **install Rasa** and some of its libraries

```
pip install rasa_nlu
```

We have successfully installed Rasa NLU, lets **install its libraries**

```
pip install rasa_nlu[spacy]
```

```
python -m spacy download en
```

```
python -m spacy download en_core_web_md
```

```
python -m spacy link en_core_web_md en
```

```
pip install rasa_nlu[tensorflow]
```

```
pip install rasa_core_sdk
```

Training the Data

These are some libraries which we use them later

Let's start training our chatbot, Before starting, let's create some folder,

```
mkdir Careerbot
```

```
cd Careerbot
```

```
mkdir fold
```

```
cd folder
mkdir data
cd data
```

Now go to data folder manually in your windows,

Now create nlu.md file

nlu.md

We are using markdown documentation for our bot, We can even use .json but I am using here markdown, which I think it as easy.

Here you can train your data using intent,entity and its examples.

To make this bot simple, I am not using any entity.Get the

nlu.md(<https://github.com/shreyas6652/Chatbot-using-RasaNLU/blob/master/CareerBot/wall-e/data/nlu.md>)

Lets create a file as nlu_config.yml in fold folder

(https://github.com/shreyas6652/Chatbot-using-RasaNLU/blob/master/CareerBot/wall-e/nlu_config.yml).

In this file we will create pipeline for our data

Now go command prompt, and train the data

```
python -m rasa_nlu.train -c nlu_config.yml --data data/nlu.md -o models
--fixed_model_name nlu --project current --verbose
```

```

Anaconda Prompt

(base) C:\Users\SHREYAS>conda activate botenv

(botenv) C:\Users\SHREYAS>cd Desktop

(botenv) C:\Users\SHREYAS\Desktop>cd CareerBot

(botenv) C:\Users\SHREYAS\Desktop\CareerBot>cd wall-e

(botenv) C:\Users\SHREYAS\Desktop\CareerBot\wall-e>python -m rasa_nlu.train -c nlu_config.yml --data data/nlu.md -o models --fixed_model_name nlu --project current --verbose
C:\Users\SHREYAS\Anaconda\envs\botenv\lib\runpy.py:125: RuntimeWarning: 'rasa_nlu.train' found in sys.modules after import of package 'rasa_nlu', but prior to execution of 'rasa_nlu.train'; this may result in unpredictable behaviour
  warn(RuntimeWarning(msg))
2019-08-08 22:53:19      rasa_nlu.utils.spacy_utils - Trying to load spacy model with name 'en'
2019-08-08 22:53:24      rasa_nlu.components - Added 'SpacyNLP' to component cache. Key 'SpacyNLP-en'.
2019-08-08 22:53:29      rasa_nlu.training_data.loading - Training data format of data/nlu.md is md
2019-08-08 22:53:29      rasa_nlu.training_data.training_data - Training data stats:
- intent examples: 1780 (8 distinct intents)
- Found intents: 'SALARY', 'GREETING', 'CAREER', 'AFFIRM', 'STUDY', 'EXAM', 'JOB_SEARCH', 'REJECT'
- entity examples: 0 (0 distinct entities)
- found entities:

2019-08-08 22:53:29      rasa_nlu.model - Starting to train component SpacyNLP
2019-08-08 22:53:42      rasa_nlu.model - Finished training component.
2019-08-08 22:53:42      rasa_nlu.model - Starting to train component SpacyTokenizer
2019-08-08 22:53:43      rasa_nlu.model - Finished training component.
2019-08-08 22:53:43      rasa_nlu.model - Starting to train component SpacyFeaturizer
2019-08-08 22:53:43      rasa_nlu.model - Finished training component.
2019-08-08 22:53:43      rasa_nlu.model - Starting to train component RegexFeaturizer
2019-08-08 22:53:43      rasa_nlu.model - Finished training component.
2019-08-08 22:53:43      rasa_nlu.model - Starting to train component CRFEntityExtractor
2019-08-08 22:53:43      rasa_nlu.model - Finished training component.
2019-08-08 22:53:43      rasa_nlu.model - Starting to train component EntitySynonymMapper
2019-08-08 22:53:43      rasa_nlu.model - Finished training component.
2019-08-08 22:53:43      rasa_nlu.model - Starting to train component SklearnIntentClassifier
Fitting 5 folds for each of 6 candidates, totalling 30 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 30 out of 30 | elapsed: 3.3min finished
2019-08-08 22:58:00      rasa_nlu.model - Finished training component.
2019-08-08 22:58:00      rasa_nlu.model - Successfully saved model into 'C:\Users\SHREYAS\Desktop\CareerBot\wall-e\models\current\nlu'
2019-08-08 22:58:00      __main__ - Finished training

```

In this you can see, We are using 1780 intent example and 8 distinct intent, even the intent name, as well entity.

After successfully training the data. Our work is to train the bot using python. We train the data in

nlu_model.py(https://github.com/shreyas6652/Chatbot-using-RasaNLU/blob/master/CareerBot/wall-e/nlu_model.py)

Save this file in the fold folder.

After training we need to run it

python nlu_model.py

Training the chatbot

After training the model, We need to create the files in the data folder. Use notepad to create (better use Notepad++)

domain.yml

stories.md

policy.yml

endpoints.yml(For local server)

action.py(To combine all file and to launch bot)

You can get all this file in

(<https://github.com/shreyas6652/Chatbot-using-RasaNLU/tree/master/CareerBot/wall-e>)

Lets us open the prompt to connect the server

python -m rasa_core_sdk.endpoint --actions actions.

Do not close the server, Open new anaconda prompt

conda activate botenv

cd CareerBot

cd fold

python -m rasa_core.train -d domain.yml -s data/stories.md -o models/dialogue -c policy.yml

After training now run the bot

python -m rasa_core.run -d models/dialogue -u models/current/nlu --endpoints endpoints.yml

```
2019-08-09 09:48:56 root - Rasa Core server is up and running on http://localhost:5005
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> hey
Hey Welcome to CareerBot, how can I help?
127.0.0.1 - - [2019-08-09 09:50:19] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 209 0.735065
Your input -> I need job
Can you Please tell your skills, We will match and send you JOBS?
127.0.0.1 - - [2019-08-09 09:50:27] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 233 0.024453
Your input -> i know python
We have noted your skills, We will soon contact you Thank you
127.0.0.1 - - [2019-08-09 09:50:34] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 229 0.025455
Your input -> thank you
Nice to hear it.
127.0.0.1 - - [2019-08-09 09:50:42] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 184 0.058872
Your input -> /stop

(botenv) C:\Users\SHREYAS\Desktop\CareerBot\wall-e>
```

Our chatbot is working successfully.

Uses of the bot

Medical applications

Health bots are designed to help with health-related issues. They give the user health related information. If we tell the symptoms, they give the related diagnosis. We can also set up an appointment and set the remainder for the doctor consultation. We can also get the suggestions for cold,cough,headache etc. We can also search for the availability of medicines in the nearby medical shops instead of visiting or calling medical shops and wasting time.

Applications in transportation

Tourism chatbots help people to search,select and book tickets. We can also ask about the good hotels and lodge facilities in the place of interest. It also gives us the information about the nearby visiting places, so that we can visit those places too. We can also book the flight tickets in advance. It can also provide the information about the cab facilities in the tourist place. Building this kind of chatbot helps us to get all the information about the particular place.

Applications in E-commerce field

Mobile messengers connected with Chatbots and the e-commerce industry business can open a new channel for selling the products online. These chatbots interact with the users and help them to either buy or sell their products efficiently online. We can also ask the information about the product based on its cost,colour,size(in case of clothes) etc. They dramatically advance the shopping experience and improve sales online.We can also get the configuration related information for the electronic gadgets which helps the customer to search efficiently.

They make our work easy and save time.

Information about the nearby multi-specialty hospitals can be asked and we can also know about the specialists for the particular disease.

Conclusion

By creating the chatbot, It reduces the man power. We get any information fast. A person may get tired of saying the same thing many times but chatbot can say the same thing many times. A person may commit mistake in giving information but chatbot never does mistake (if the person has entered the data correctly in bot). We need to pay for person every month, But chatbot is one time investment.

Advantages

- Gives information rapidly
- Makes less error
- Does Not require human beings
- Available anytime
- User friendly
- Bias information

Disadvantages

- Makes the same kind of mistakes always
- Needs to update data now and then.
- People are not comfortable with chatbot
- Trusting a chatbot is not easy
- ChatBot may get hack, So need to provide proper security
- Old people can't adjust to the new system easily.

Thank you