

Rasa NLU Chatbot

Business Objective

In simple terms, a chatbot is a computer program that simulates and processes human conversation, either written or spoken, allowing humans to interact with digital devices as if they were communicating with a real person.

Chatbots can be as simple as a program that can answer a simple query with a single-line response or as sophisticated as digital assistants that learn and evolve to deliver increasing levels of personalization as they gather and process information.

There are two main types of chatbots:

➤ Rule-based chatbots

In a rule-based type of bot, communication is through a pre-defined rule. User input must conform to these pre-set rules to get an answer.

Example: A chatbot that answers basic customer inquiries like the status of one's delivery, return request process, tracking information, etc.

➤ AI-based chatbots

AI chatbots make use machine learning to understand the context and intent of a question before formulating a response. These chatbots generate their own answers to more complicated questions using natural-language responses. The more you use and train these bots, the more they learn and the better they operate with the user.

The working of an AI-driven chatbot depends on two main concepts, the intent and the entity.

- Intent - Intention or purpose of the user in the conversational flow.
- Entity - A data point or value which you can extract from a conversation

In our case study, we will use the Rasa NLU model to build the chatbot.

Rasa is an open-source machine learning framework for automated text and voice-based conversations. Understand messages, hold conversations, and connect to messaging channels and APIs.

We will build an AI-based chatbot using an E-Commerce business case.

Data Description

The data can be curated from the following two data curation websites,

- Rasa NLU trainer: <https://rasahq.github.io/rasa-nlu-trainer/>
- Chatito: <https://rodrigopivi.github.io/Chatito/>
 - The business case we will be considering is E-Commerce,
 - The intents consist of: product_info, ask_price, cancel_order
 - The entities consist of: product, location and order_id

Aim

To build an AI chatbot using the Rasa NLU model.

Tech stack

Language - Python

Libraries – pandas, matplotlib, rasa, pymongo, tensorflow, spacy

Approach

1. Perform data curation to obtain the data
2. Importing the required packages and libraries
3. Import the data
4. Create a function to convert the data into training and testing dataframes
5. Convert the dataframes to JSON files
6. Exploratory Data Analysis (EDA) –
 - Data Visualization
7. Create configuration files (.yaml) for spacy and TensorFlow.
8. Model Building
 - Define a function to train the Rasa NLU model.
9. Model Evaluation
 - Define a function to perform a model evaluation on test data
10. Train the data using spacy as the pipeline
11. Train the data using TensorFlow as the pipeline
12. Plot confusion matrix for both the models
13. Interpreting the model
14. Install MongoDB and import pymongo
15. Create an IntentFlow class
16. Create a ContextManager class

17. Create a function for processing a message.
18. Test the chatbot

Modular code overview

```
input
|_data.json
|_spacy_config.yml
|_Tensorflow.config.yml

src
|_Engine.py
|_ML_Pipeline
    |_db.py
    |_dialog.py
    |_infer.py
    |_training.py
    |_utlis.py

lib
|_rasa_chatbot.ipynb

output
|_spacy_model
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. Input
2. Src
3. Output
4. Lib

1. Input folder - It contains all the data that we have for analysis. There are three json files along with two configuration files present.
 - data.json
 - spacy_config.yml
 - Tensorflow_config.yml
2. Src folder - This is the most important folder of the project. This folder contains all the modularized code for all the above steps in a modularized manner. This folder consists of:
 - Engine.py
 - ML_Pipeline

The ML_Pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file.

3. Output folder – The output folder contains the best fitted model that we trained for this data. This model can be easily loaded and used for future use and the user need not have to train all the models from the beginning.

Note: This model is built over a chunk of data. One can obtain the model for the entire data by running engine.py by taking the entire data to train the models.

4. Lib folder - This is a reference folder. It contains the original lpython notebook that we saw in the videos. The reference ppt used is also included in this folder.

Project Takeaways

1. Understanding the business problem.
2. Understanding different types of chatbots.
3. Understanding the concept of intent and entities.
4. Performing data curation
5. Understanding the Rasa NLU model
6. Understanding configuration needed for spacy and TensorFlow
7. Importing the dataset and required libraries.
8. Creating train and test data files
9. Performing basic Exploratory Data Analysis (EDA).
10. Learn how to train the Rasa NLU model with spacy and TensorFlow as the pipelines.
11. Evaluate these models by plotting the confusion matrix and choose the appropriate model
12. Interpret data using the model
13. Understanding the installation and use of MongoDB.
14. Learn how to create a class function for the intent flow and context manager
15. Create a function for processing the message
16. Finally, testing the working of Rasa NLU chatbot