**Chatbot**

A chatbot is an artificial intelligence (AI) software that can simulate a conversation or a chat with a user in natural language through messaging applications, websites, mobile apps or through the telephone. A chatbot is also known as **chat robot**, talk bot, **chatterbot** or **conversational agents.** At the heart of chatbot technology lies **NLP** (Natural language processing), that forms the basis of the voice recognition systems used by virtual assistants such as Google assistant, Apple’s Siri, and Microsoft’s Cortana. We can use both text and speech to communicate with it.

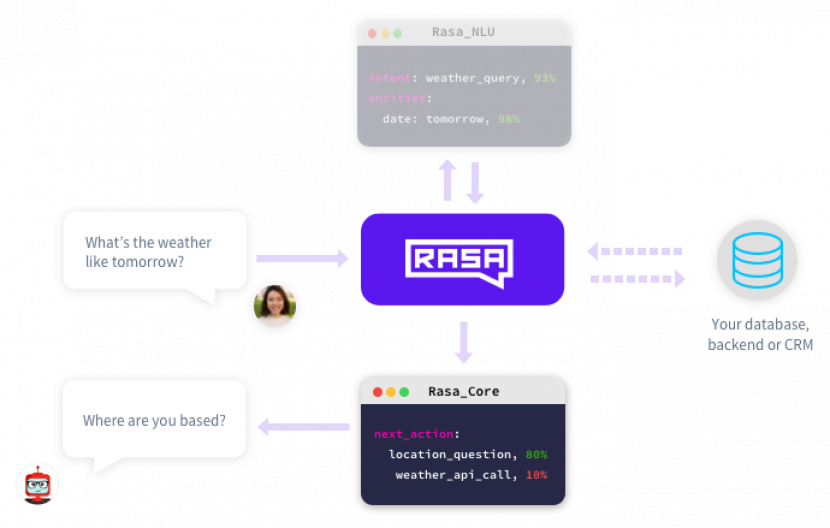
**What is RASA?**

Chatbots are of two types, rule-based and self-learning bots. Rule-based bots fails to manage complex ones and self-learning bots are mainly programmed by LSTM(Time series). RASA framework gives highly sophisticated NLP solutions. RASA is a framework to create a chatbot without any programming language. RASA is an open source AI tool which can be easily installed in local machines.

RASA consists of two components,

**RASA NLU:** It is a natural language processing tool for classification of intent and extraction of entity from the user input and it helps the bot to understand the words of the user.

**RASA Core:** It’s a chatbot framework with machine learning-based dialogue management which takes the structured input from the NLU and predicts the **next best action** using a probabilistic model like LSTM neural network.



Pic credits: chatbotslife.com

From the above picture we can understand that RASA NLU classifies intents and entities and RASA core is responsible for the action to be performed

**Some basic definitions**

·      **Intent:** Intent is nothing but what the user is aiming for. Greetings like hi, hello can be considered as intent.

·      **Entity:** Entity is the extraction of useful information from the user input.

·      **Stories:** It defines the sample interaction between user and the chatbot.

·      **Actions:** They are basically the operations performed by the bot either asking some details to get all entities or querying database to get some information.

·      **Slots:**  Slots are like placeholders for the values that enable the bot to keep the track of the conversations.

·      **Templates:**  It is the response from the bot for the user input.

·      **Pipeline:**In RASA NLU, incoming messages are executed one after the other through pipeline. They are the components for intent classification and entity extraction.

**Installation and setup**

First we have to set up a virtual environment. I’m using anaconda prompt for setting up virtual environment. Prerequisite software for the installation is

Microsoft Visual Studio(<https://visualstudio.microsoft.com/>).

      First we have to activate conda by,

Ø conda activate botenv

      To deactivate,

Ø conda deactivate

 To install RASA,

Ø pip install rasa\_nlu or Python3 install rasa\_nlu

      To install its libraries,

Ø pip install rasa\_nlu[spacy]

Ø python3 -m spacy download en

Ø python3 -m spacy download en\_core\_web\_md

Ø python3 -m spacy link en\_core\_web\_md en

Ø pip install rasa\_nlu[tensorflow]

**Training the data**

For training data files, we create data directory under bot and create the training file nlu.md in that.

mkdir bot

 cd bot

 mkdir  data

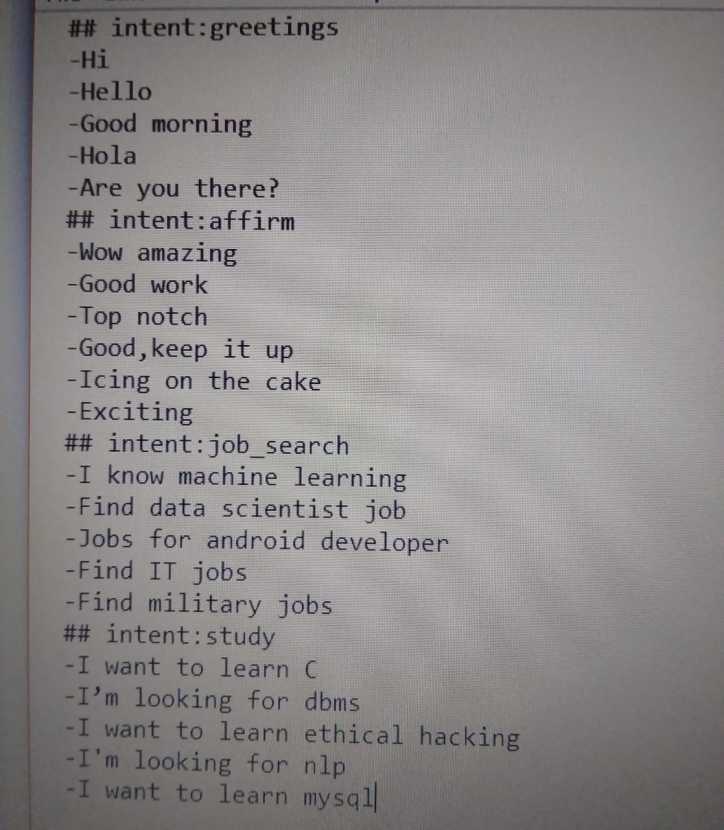
 cd  data

 touch  nlu.md

Now go to command prompt and train the data

python3 -m rasa\_nlu.train -c nlu\_config.yml --data data/nlu.md -o models --fixed\_model\_name nlu --project current –verbose

Now, train the bot by the following intents and entities



and so on….

“## intent” defines the name of the intent and the word defines entity.

Since we are using python, create nlu\_model.py in the base project directory (i.e. bot).

After training the data run it by the command,

Ø python nlu\_model.py

**Training the chatbot**

After training the data we create the files,

domain.yml, stories.md, policy.yml , endpoints.yml, action.py

Connect to the server by,

python -m rasa\_core\_sdk.endpoint --actions actions

conda activate botenv

cd bot

cd data

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

 Now run the bot by,

python -m rasa\_core.run -d models/dialogue -u models/current/nlu --endpoints endpoints.yml

**Now the chatbot is working successfully.**

**Conclusion**

Finally, we have successfully completed the training of career bot which understands the natural language. We can add more data so that the bot understands better. They save both time and money. Since chatbot is a onetime investment it saves money rather than hiring workers. They provide customer satisfaction and chances of errors are less(unless the user gives correct information). It is always user friendly.