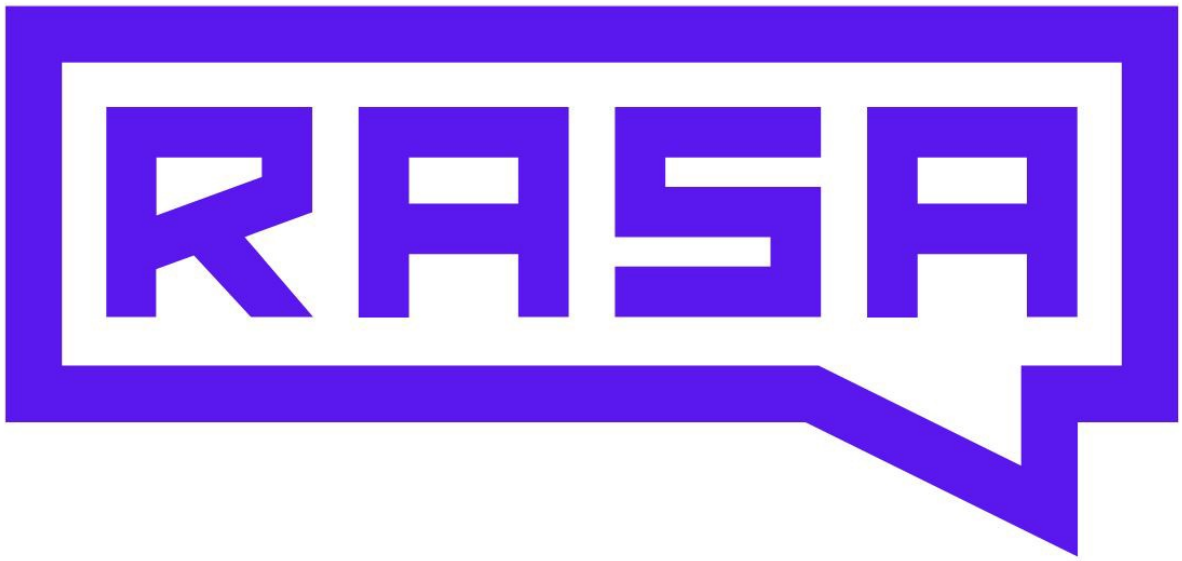

Look into Rasa

26-05-2021

Levi Nauwelaerts
BlueSky Unlimited

Overview

I looked into using the platform Rasa instead of using our current plain python model, with the goal to see if this would be more developer friendly or not. I got some helpful insights in it.

Goals

1. Is the rasa platform an easier tool ?
2. What are the benefits and what are the cons?

Specifications

On a windows 10 machine

- Latest version of anaconda
- Microsoft Visual C++ Redistributable
- Python 3.7.6
- ujson 2.0.3
- Tensorflow 2.1.0
- rasa 2.2.5 (free version)

Setup process

Step 1

Install the latest version of anaconda, this will act as a closed off python environment but also ease the transfer of state between developers.

Step 2

Download Microsoft Visual C++ Redistributable.

Step 3

Create a folder on your machine for your project.

Step 4

Open anaconda and in anaconda cd to the path of our new folder.

Step 5

In anaconda run the following command:

conda create --name myChatbot python==3.7.6

Where the name “myChatbot” can be any name for our chatbot environment.

Step 6

In anaconda run the following command:

conda activate myChatbot

activating the env to work in.

Step 7

In anaconda run the following command:

conda install ujson==2.0.3

Step 8

In anaconda run the following command:

conda install tensorflow==2.1.0

Step 9

In anaconda run the following command:

pip install rasa==2.2.5

Step 10

In anaconda run the following command:

rasa init

This can take a while.

Verify the project location and confirm.

Train the model. This again will take a while.

Working with the chatbot

Start rasa

In our anaconda environment, run the following command:

rasa shell

This will start the model, this will take a while, after this you can in the console speak to the chatbot.

To exit the progress run the /stop command.

Deploying on a webpage

In essence it is just adding the script to the html which has the websocket for the chatbot.

Also edit the file credentials.yml uncomment the socketio entry and edit its information.

```
Socketio:
    user_message_evt: user_uttered
    bot_message_evt: bot_uttered
    session_persistence: true
```

Comparing back

Compared to our own pyhton model based on the guide of "tech with Tim".

While Rasa would be easier for those who want a chatbot and do not understand any of the Ai process, we found actually having more control over our own model has more benefits. Rasa is pretty easy and fast to set up making it ideal for a neet, however altering to exactly what you need any further than the intends is more of a black box approach.

The guide we followed for the plain python model was actually pretty clear and leaves a lot of room open for altering the logic of the model to your own needs, while in rasa this can prove more of a challenge. But setting up rasa in anaconda showed us the strength of anaconda as a development tool, being able to export our own little python "virtual machines" avoid a lot of the hassle of making sure all path and such are correctly set for each developer. And for our plain model even though the stemmer often split up the strings into small pieces they seemed to make more sense to me in the end, if they however always produce a more sensible results is harder to test and prove given the nature of the neural networks and their inputs / training sets.

My verdict

If your team needs an out of the box solution, and may be willing to spend some cash to its upkeep and improvements than rase and its enterprise line of subscription might be the way to go. However if your team / product can utilize the freedom or even need it I think going with your own plain model and then including what in essence are the same libraries will result in a better specialized chatbot for your needs.

In the case of BlueSky we went with our own model, which allowed us for a better integration to the systems we had already developed, and a more hands on approach for our developers made it easier to try to build the tools we need.

Sources

I followed the guide: "tech with Tim" for the plain python model and

took part in an udemy course for rasa: