

EPITA

Natural Language Processing

AIS

Course 5

Chatbot implementation with Rasa framework

Preamble

rasa installation (see <https://rasa.com/docs/rasa/installation/> for more details)

You can use the command : ***pip3 install rasa*** (or ***pip install rasa***)

Once it is done you can use the command: ***rasa init***
in order to create a first project to start to work with.

The initial project already has some basic stories, and intent training data enough, to train a chatbot from it. Go in the project folder and use the command ***rasa train***.

Then, you can start to interact with the chatbot for (very) basics talks using the commands:
rasa shell (if you want to talk with it directly into the command line)
rasa x (if you want to talk with it from the local Rasa X mode into a web browser).

Important: *if you do not want or cannot install Rasa, **do not hesitate** to use instead <https://rasa.com/docs/rasa/playground/> which will allow you to test, and modify, a simple chatbot **without installing anything**. It may not be robust enough to complete the whole exercise but I should enough to complete the first few questions.*

Advice 1

For a complete list of rasa command line, do not hesitate to have a look to: <https://rasa.com/docs/rasa/command-line-interface/>

Advice 2

When you work on a Rasa chatbot, it is a good practice to always have open in an editor (for example Pycharm) the following files:

- ***nlu.py***
- ***domain.yml***
- ***rules.yml***
- ***stories.yml***

since each new feature that you will want to add will need you to modify (at least !) the four files.

Advice 3

Try to train your model each time you change something, unless it will be hard to debug.

Advice 4

Please do not hesitate to have a look at the documentation *rasa_ebook.pdf* given in the shared Teams space

Exercise

Question 1: Train the chatbot from the initial project and start a conversation with it (see the instructions above). Make sure the already defined stories like “happy path” or “sad path 1” are effective.

Question 2: Add a “ask_for_restaurant” intent, a “ask_for_hostel” intent and a “ask_for_train_ticket” intent. Make sure to use enough examples for each intent to make the chatbot able to generalize.

Define a specific story for each and make sure the chatbot is able to understand each intents

Do not forget to train the rasa model each time you add an intent or a story.

Hint: you can have a look to the *epita_rasa* project to have a rough idea of how to define “ask_for_restaurant” intent and the specific story

Question 3: Make the model more complex. When the chatbot manages to understand you want to find a restaurant, define an action from it asking which kind of food you want.

Hint: you may need to hardcode this interaction using rules (see *rules.yml*)

Question 4: Define three more intents, “ask_for_italian_food”, “ask_for_french_food”, “ask_for_chinese_food”, and complete the story to allow a complete dialogue with the chatbot

Question 5: Make the conversation more open by stopping restricting to only three kind of food but make it completely generic using *slot* concept

Hint: you may have a look at the *epita_rasa* project and see how the restaurant slot is defined. You may have a look as well at the *ask_for_specific_restaurant* intent, to understand how to use *slot* into an intent.

Question 6: Design a complex intent with two slots in it. One for the restaurant and the other one for the province. The chatbot should be able to understand both, and to give you a reply with the address of the restaurant in the province wanted. Take this information from the *FastFoodRestaurant.csv* file given.

Hint: you will probably have to use custom actions (*actions.py* file). Do not hesitate to have a look at the documentation *rasa_ebook.pdf*