

**Conversational Chatbots**

*AMOS Project WS 2017/18 - Technical University Berlin in cooperation with Actano GmbH*

Documentation

### Table of Contents

Page #

1. Introduction and Objective 3
2. User Manual 5
   1. Pre-requisites 5
   2. Getting Started 6
   3. How to use the bot 7
3. Architecture 8
4. Used Technologies and Tools
   1. RASA Core
   2. RASA NLU
   3. Workstreams
   4. Docker
   5. Amazon Web Services

**1. Introduction and Objective**

This Conversational Chatbot project was part of the AMOS project course at the Technical University of Berlin. The project was carried out in cooperation with Actano GmbH. The aim of the course is to make software development process more flexible and smoother to participants, by following agile software development methods, particularly using scrum. In addition, as the course aims to provide participants with real world conditions facing software development processes, the projects carried out in the course are mostly offered from industrial partners, which also apply scrum in their working environments.

The main objective of the developed conversational bot was to deliver a smalltalk, in alignment with chat bots and the RASA framework, which is based on natural language processing, in order to inform the user about agile software development methods, mainly Scrum and Kanban. The development of the conversational bot has involved using a variety of technologies and tools, which can be classified as follows:

* **RASA NLU:** An open source tool used for intent classification and entity extraction. Mainly used for training available data based on probabilistic models.
* **RASA Core:** An open framework that is used in constructing bots. A bot is composed of a domain file, which specifies allowed actions, intents and entities, as well as stories that define the conversation flow of the bot. Based on the trained data in the RASA NLU, RASA core gives a response, upon sending a message (a request), as long as it is recognized.
* **Amazon Web Services:** Deploying the bot to be used through an API by Actano, the industrial partner of the project.
* **Travis:** Testing code that is committed to the GitHub repository, as well as continuous integration, and deployment.
* **Python:** Programming language that has been used in order to define various features regarding server/client connections (HTTP-service), session handling, and ensuring correct functionality and behavior of the conversational bot.
* **Docker:** Running the conversational bot through HTTP-service.
* **Redis DB:** Storing session data, so that the bot can recognize conversation dialogue context.

As a frontend for this conversational bots, Workstreams on Slack was used. Actano has provided the Team with an API, which was capable of conversing and communicating with the bot.

The github repository for this project can be found at: https://github.com/amos-ws17/amos-ws17-proj1

The Team consisted of 7 contributors, which have developed the conversational bot in a total of 13 sprints, each sprint lasting one week. Every team member has at least played the role of being a product owner or a scrum master at least one time, while for the rest of the weeks contributing as a software developer.

This documentation aims to give an overview at the pre-requisites that you need to take into consideration, if you were interested in using the conversational chatbot, besides the architecture on which the bot is operating, and the role of each used technology during development in more detail.

**2. User Manual**

If you would like to use the conversational bots, please follow the instructions that are provided by this section.

**a. Pre-requisites**

**i) Python - version 3.6.**

For more Information, please refer to: https://www.python.org/downloads/release/python-363/

**ii) venv - Virtual Environment for executing Python code**

Install a virtualenv in a desired location

sudo pip install virtualenv

# cd to our desired location

virtualenv -p python venv

Start your venv

source venv/bin/activate

While you are still in the terminal, you may deactivate venv

deactivate

**iii) Install spacy, scikit-learn, flask and pytest - Python libraries**

pip install -U spacy

python -m spacy download en

pip install -U scikit-learn scipy sklearn-crfsuite

pip install flask

pip install -U pytest

**iv) Install RASA Framework**

Install Rasa NLU (for more Information, please refer to: http://rasa-nlu.readthedocs.io/en/latest/installation.html)

pip install rasa\_nlu

Install Rasa Core (for more Information, please refer to: https://core.rasa.ai/installation.html)

pip install rasa\_core

**b. Getting Started**

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes. See deployment for notes on how to deploy the project on a live system.

### *Deployment*

To deploy the system make sure you have docker and git installed. If you do, clone this repository with:

git clone https://github.com/amos-ws17/amos-ws17-proj1.git

Then navigate to the newly created amos-ws17-proj1 folder and type in the following command to start deploy the bots:

docker-compose up

### Train and run the service

*Start your venv*

# cd to our desired location

source venv/bin/activate

*Train the models*

# cd to amos-ws17/workstreambot

python -m train -d scrum+kanban -n nlu\_training\_data\_full.json

*Run the service*

python -m http\_service -d scrum+kanban

*Start your venv*

# cd to our desired location

source venv/bin/activate

*Train the models*

# cd to amos-ws17/workstreambot

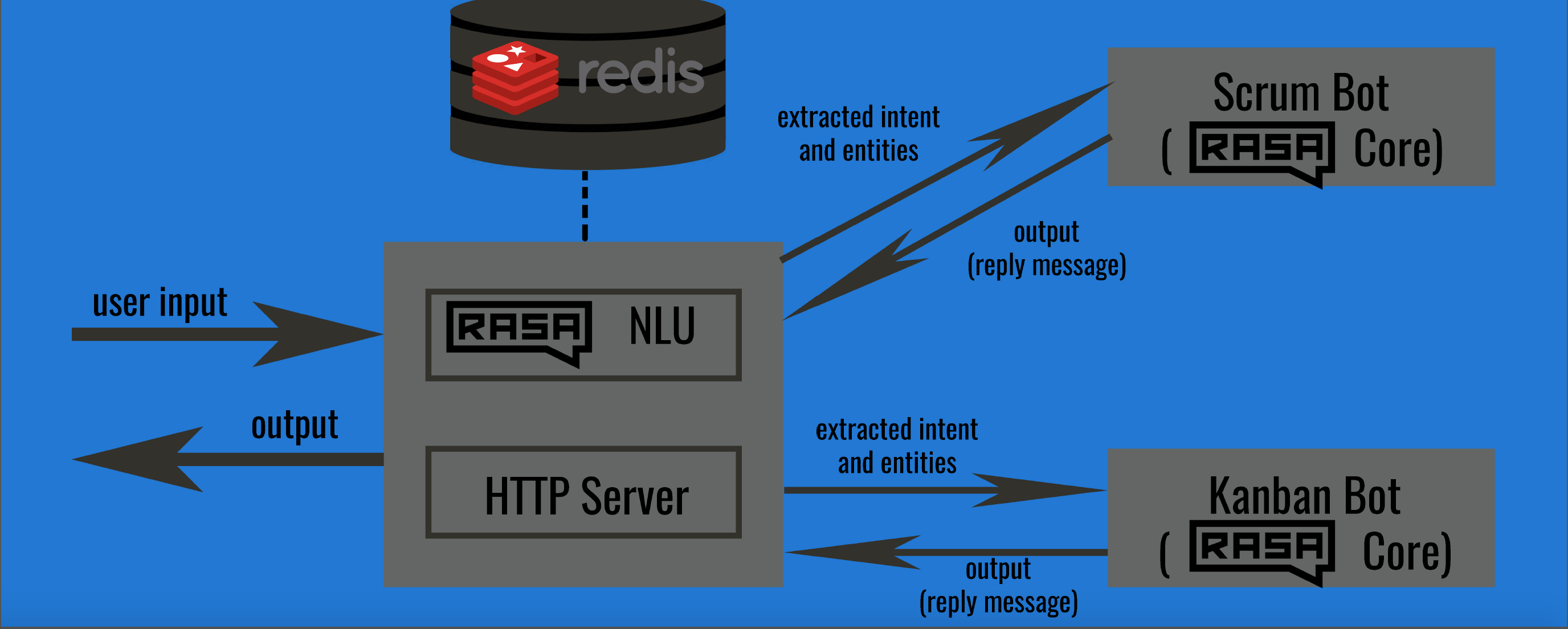
python -m train -d scrum+kanban -n nlu\_training\_data\_full.json

*Run the tests*

python -m pytest test/ # execute all component tests

**c. How to use the bot**

**3. Architecture**

****