RASA NLU Documentation

Sikim Chakraborty

Contents

1 INSTALLATION

1.1 PYTHON

OS: Linux (Ubuntu 16.04 LTS)
Download Anaconda distribution of Python
URL: **https://www.continuum.io/downloads**
Version: Python 2.7 (64 bit)

After downloading transfer Anaconda setup file to VM using an FTP client such as FileZilla.
After installation of Anaconda, restart VM

Update Conda:

```
conda update conda
```

Update Packages:

```
conda update --all
```

1.2 RASA NLU

1.2.1 PREREQUISITES

GIT:

```
conda install git
```

NumPy:

```
pip install -U numpy
```

Scikit-Learn:
Installed with Anaconda distribution of Python

1.2.2 INSTALLATION

Install RASA NLU:.[1]

```
pip install rasa_nlu
```

Bleeding Edge Version:

```
pip install git+https://github.com/golastmile/rasa_nlu
```

1.2.3 SETTING UP A BACKEND

spaCy + sklearn:

```
pip install -U spacy
```

Bleeding Edge Version:.[2]

```
pip install git+https://github.com/explosion/spaCy
```

Download English Model(Dictionary):

```
python -m spacy download en
```

1.2.4 RASA NLU TRAINER

Installation:.[3]

```
npm i -g rasa-nlu-trainer
```

[1] If NumPy installation conflicts with pre existisin NumPy installation uninstall using: **pip uninstall numpy**

[2] ImportError: No module named builtins use: **pip install future**

[3] Prerequisites: **nodejs** and **npm**

2 USAGE

2.1 TRAINING THE MODEL

First create a directory for Project:

Example:

```
mkdir RNLU
```

Create a config file within **RNLU**:

```
cd RNLU
sudo nano json.config
```

```
{
  "backend": "spacy_sklearn",
  "path" : "./models",
  "data" : "./data/examples/rasa/mem-rasa.json"
}
```

```
cntrl + o
cntrl + x
```

2.1.1 CREATING TRAINING EXAMPLES FOR RASA NLU:

Make a directory called **data** within **RNLU**:

```
mkdir data
```

Make a directory called **examples** within **data**

```
cd data
mkdir examples
```

Make a directory called **rasa** within **examples**

```
cd examples
mkdir rasa
```

Create training data json file:

Example for Mem+

```
sudo nano mem-rasa.json
```

```
{
  "rasa_nlu_data": {
    "common_examples": [
      {
        "text": "hey",
        "intent": "greet",
        "entities": []
      },
      {
        "text": "i want to see 200 george street equipment view",
        "intent": "navigation",
        "entities": [
          {
            "start": 14,
            "end": 31,
            "value": "200 george street",
            "entity": "building_name"
          },
          {
            "start": 32,
            "end": 41,
            "value": "equipment",
            "entity": "target"
          }
        }
    ]
  }
}
```

2.1.2 VISUALIZING THE TRAINING DATA:

This can only be done in Windows since I used an Ubuntu Server VM. Training data can also be prepared using this method by clicking on the add new example button and adding the Utterances followed by their respective Intents and Entities(which can be selected from the utterances). After preparing the training data, the ***mem-rasa.json*** file can be transferred to the VM using FileZilla.

Running the RASA NLU TRAINER:

```
cd RNLU/data/examples/rasa
rasa-nlu-trainer
```

2.1.3 TRAINING THE MODEL USING SPACY AND SKLEARN:

Navigate to the project folder i.e. **RNLU(cd RNLU)**:

```
python -m rasa_nlu.train -c config.json
```

A folder called **models** will be created bearing the time stamp at which it was trained. Any subsequently trained models will be added in the said folder and the model to be referrenced has to changed at the time of hosting the server, in the **Metadata.load()** method.

2.2 TESTING THE MODEL

2.2.1 CREATING A TEXT FILE WITH SAMPLE UTTERANCES OR TEST UTTERANCES

Example:

rnlu_test.txt

show me 200 george street overview
take me to 200 george street overview page
i want to see the overview page of 200 george street
take me to 200 george street energy page
take me to 200 george street equipment page
show me the equipment page of 200 george street
i want to see the energy page of 200 george street

2.2.2 PYTHON SCRIPT TO CHECK TEST UTTERANCES OUTPUT

Example:

rnlu_test.py

```
from rasa_nlu.model import Metadata, Interpreter
from rasa_nlu.config import RasaNLUConfig
import json

metadata = Metadata.load('./models/model_20170403-084112')
interpreter = Interpreter.load(metadata, RasaNLUConfig("config.json"))

with open("test_data.txt","r") as f:
        t_data = f.readlines()

t_data =[x.strip() for x in t_data]
output = []
for st in t_data:
        resp = interpreter.parse(unicode(st))
        output.append(resp)

f1 = open('test_output.txt', 'w')
f1.write(json.dumps(output, indent=4))
f1.close()
```

The output will be available in the text file **test_output.txt**. It will comprises of the the **utterances** with their respective **Intent** and **Entity** classification and other parameters such as **confidence score** in the **.json** format

3 RUNNING THE RASA NLU SERVER FROM VM

3.0.1 CONNECT TO THE VM USING PUTTY

Host name(or IP Address): 52.170.40.190
Port: 22
Connection Type: SSH

Username: *****
Password: ******

3.0.2 RUNNING THE SERVER:

Starting the server by running the Python script

```
cd RNLU
cd fRNLU
python fRNLU.py
```

3.0.3 ACCESSING THE SERVICES

Base URL: http://52.170.40.190:5000
Port No: 5000

Endpoint URL: http://52.170.40.190:5000/fetchentityintent/<message> Sample Output:

URL Used: http://52.170.40.190:5000/fetchentityintent/hi

Sample Output:


```
{
  "entities": [],
  "intent": {
    "confidence": 0.9396765467822804,
    "name": "greet"
  },
  "intent_ranking": [
    {
      "confidence": 0.9396765467822804,
      "name": "greet"
    },
    {
      "confidence": 0.06032345321771961,
      "name": "navigation"
    }
  ],
  "text": "hi"
}
```

Postman Screenshot: