

Natural Language Processing

Report for Assignment (Rasa Chatbot)

Vikanksh Nath (MT19AI024)

Objective: We are required to do the following tasks

1. Set up Rasa
2. Run the restaurant chatbot
3. Customize the restaurant chatbot to handle queries for your own neighborhood - data set preparation
4. Add "feedback" intent to the pre-existing chatbot - the idea being to converse with the user to get feedback about an earlier restaurant that was suggested by the chatbot

Task 1: As we are at home, we found Google-Colab convenient to use. As to train a rasa core model, high computational power is required. So, we directly used colab to install both rasa nlu, core and all related dependencies. All are shown in cells of the submitted single colab notebook.

Task 2: For our Restaurant Chatbot we first created nlu.md file. It contains the following intents.

- affirm : yes, yup etc.
- ask_budget : to indicate budget is told by user
- ask_restaurant : user is asking about restaurant
- bye : user is saying bye to bot

- deny : to indicate denial by user
- goodbye : to indicate bye by user
- greet : to indicate greet by user
- out_of_scope : to handle some out of scope flow of conversation
- thank : to indicate thank by user
- feedback : to indicate feedback by user
- feed_resturantName : to indicate restaurant name is given by user for feedback

▼ Part 0: setup workspace

Let's start with jupyter configuration

```
[1] %matplotlib inline

import logging, io, json, warnings
logging.basicConfig(level="INFO")
warnings.filterwarnings('ignore')

def pprint(o):
    # small helper to make dict dumps a bit prettier
    print(json.dumps(o, indent=2))
```

▼ Install Rasa NLU and Rasa Core. You can jump this step if you have already installed it on your PC.

```
[2] import sys
python = sys.executable

# In your environment run:
!{python} -m pip install -U rasa_core==0.9.6 rasa_nlu[spacy]==0.12.3;

# as well as install a language model:
!{python} -m spacy download en_core_web_md
!{python} -m spacy link en_core_web_md en --force;
```

Figure 1. Snap of Rasa Setup from Notebook

We also created following entities and same slots : - budget, - cuisine, - restaurant, - location and lookup table for location, restaurant and cuisines also. We also made synonyms for required entries.

Note: Location lookup table, synonyms and training example for nlu.md is taken from the internet. But details of our neighborhood : 'Muzaffar Nagar' are added after.

We tested our trained nlu model with a sentence, corresponding output is shown in the notebook as well as in snaps here.

```
pprint(interpreter.parse("I want a chinese restaurant in Muzaffarnagar"))  
  
  "confidence": 0.6681254552575617,  
  "extractor": "ner_crf",  
  "processors": [  
    "ner_synonyms"  
  ]  
},  
{  
  "start": 30,  
  "end": 43,  
  "value": "muzaffarnagar",  
  "entity": "location",  
  "confidence": 0.9891883449305706,  
  "extractor": "ner_crf"  
}  
],  
"intent_ranking": [  
  {  
    "name": "ask_restaurant",  
    "confidence": 0.7883903394901108  
  },  
  {  
    "name": "out_of_scope",  
    "confidence": 0.18419634951087957  
  },  
  {  
    "name": "feed_restaurantName",  
    "confidence": 0.005627675132806416  
  }  
]
```

Figure 2. Snap of trained nlu output showing correct intent and entity from Notebook

	precision	recall	f1-score	support
affirm	0.98	0.95	0.96	55
ask_budget	1.00	1.00	1.00	3
ask_restaurant	0.94	0.98	0.96	255
bye	1.00	0.96	0.98	23
deny	1.00	0.98	0.99	43
feed_resturantName	1.00	0.15	0.27	13
feedback	0.97	1.00	0.98	29
greet	0.98	0.96	0.97	67
out_of_scope	0.95	0.99	0.97	109
thank	1.00	1.00	1.00	26
avg / total	0.96	0.96	0.95	623

Figure 3. Snap of Intents Scores from Notebook

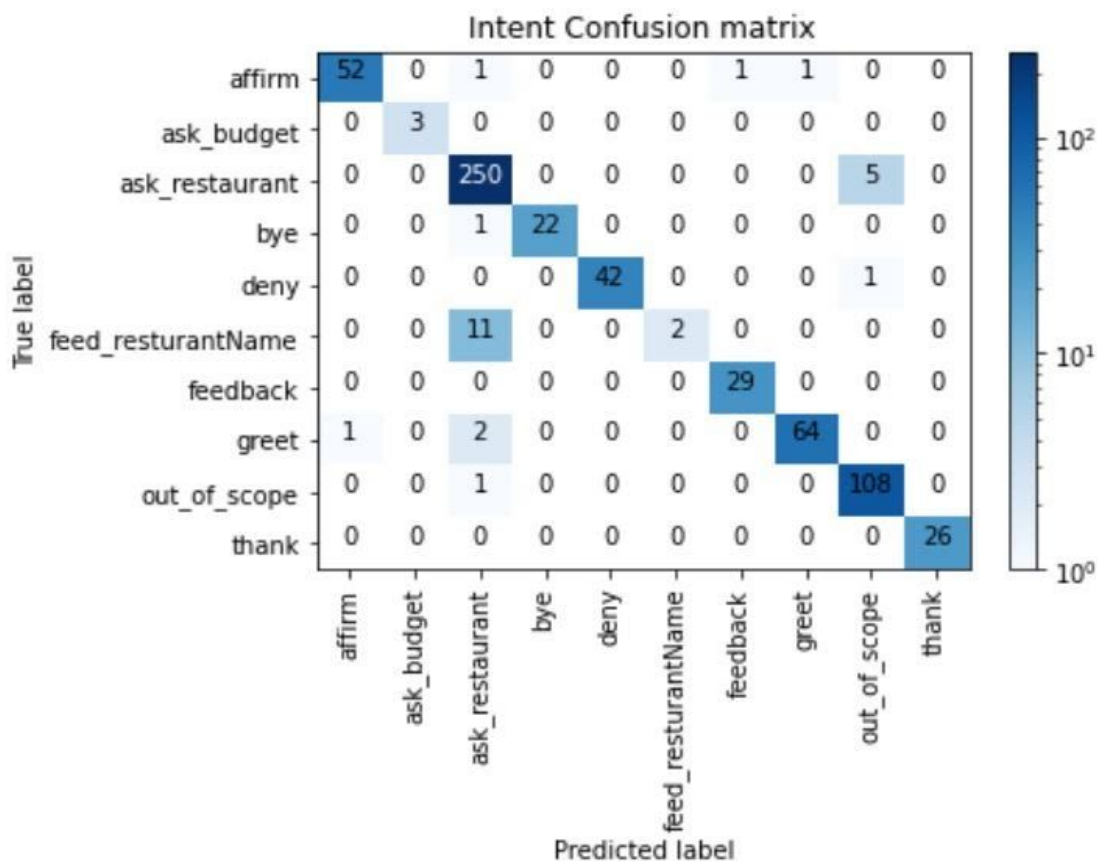


Figure 4. Snap of Intent Confusion Matrix from Notebook

We used only utter_actions. Details are shown in a notebook in the cell containing the domain.yml file.

Task 3: We created an extensive data set of *122 stories*, to handle different conversational flows, and in location the Muzaffar Nagar is used in most of the stories as a valid location. Moreover the details of local/pre-fed restaurants can only be retrieved as an output as only utter_actions are used. One may use a third-party api as zomato, to retrieve names of restaurants asked by the user for a location.

Task 4: We defined intent feedback which indicates the rating of a restaurant is given by the user. One may see the *typical flow of a story in below Fig 5*, for better understanding. First bot asks the user if, bot has assisted user before using action utter_taken_help, then if user gives a restaurant name, bot recognise it using intent : feed_resturantName then bot asks to give rating by calling action utter_feedback, and corresponding user input is recognised by bot via intent : feedback.

In this way we tried to take the user feedback for visited restaurants.

We used the colab TPU to train the Rasa Core Model, and our validation accuracy was 78-79 %.

Moreover a typical conversation of our chatbot is shown in below Fig 6.

```
## story_92_location_valid_feedback
* ask_restaurant
  - utter_ask_location
* ask_restaurant{"location": "delhi"}
  - utter_location_invalid
  - utter_ask_location_retry
* affirm
  - utter_ask_location
* ask_restaurant{"location": "Bangalore"}
  - utter_ask_cuisine
* ask_restaurant{"cuisine": "South Indian"}
  - utter_ask_budget
* ask_budget{"budget": "300"}
  - utter_ask_details
* out_of_scope
  - utter_ask_details
* affirm
  - utter_south_list
* thank
  - utter_taken_help
* feed_resturantName{"resturant": "Sangam"}
  - utter_feedback
* feedback
  - utter_happy
* bye
  - utter_bye
```

Figure 5. Snap of a story from Notebook

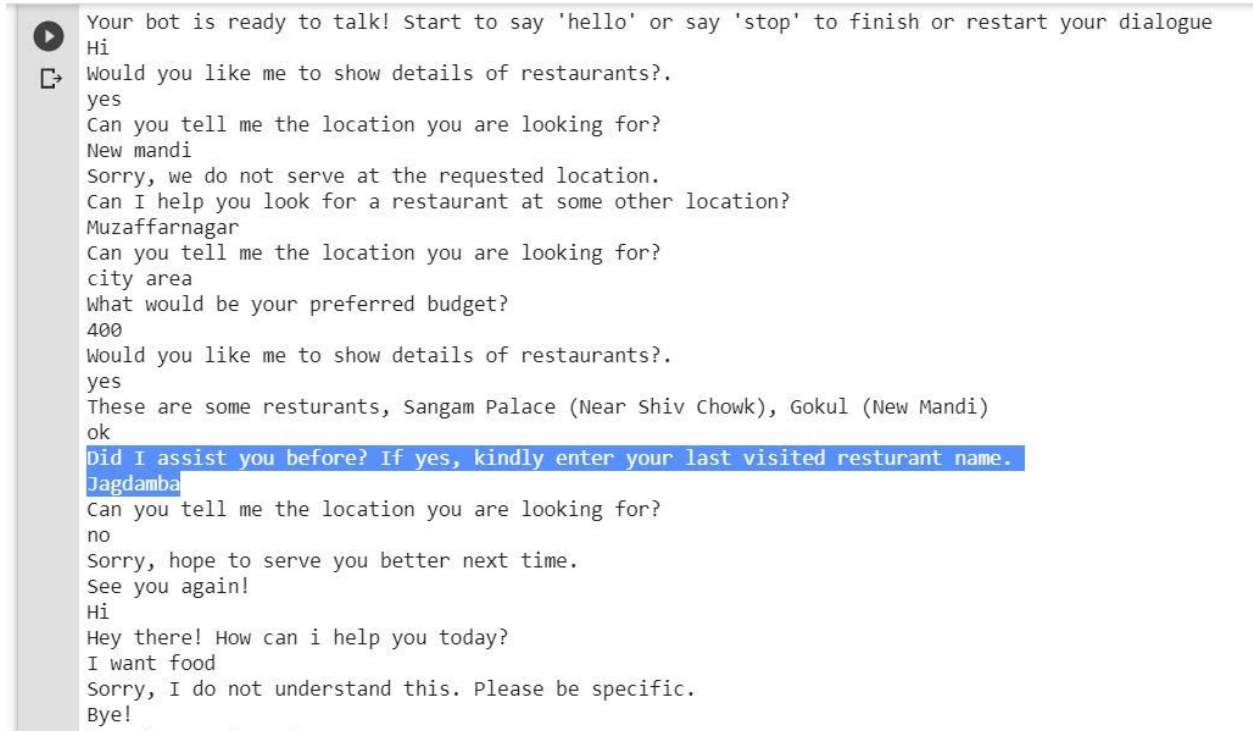


Figure 5. Snap of a Conversation with chatbot from Notebook