Rasa

* Domain.yml → The domain file is a directory of everything your assitant knows

Responses – These are the things assitant can say to users.

Intents → these are the caterires of things users say
(these are the classes for multiclass classifications).

Slots – These are the variables remembered over the course of a conversation.
(similar to f-string in python)

Entities – These are the pieces of info extracted from incoming text

forms & actions – These add application logic and extend what your assitant can do.

* Data

- The text data used to pretrain any models or features you are using (e.g. language models, word embedding, etc).

- User generated text

- Patterns of conversation

e.g. customer support logs (Assuming data

collection and reuse is covered in your privacy policy).

- User conversation with your assistant.

Data folder contains three files.
1) stories.yml 2) rules.yml 3) nlu.yml

1) Stories — training data to teach your assistant what it should do next.

2) Rules — a way to describe short pieces of conversations that always go the same way.

3) nlu —

Intents → intents are classes for multiclass classification

Why fewer intents?

Page No.:
Date

* Entities → Entities are structured pieces of information inside of a user message.

An entity can be any important detail your assistant could use later in conversation
e.g. Numbers, Dates, country names, product names, etc.

Training data for entity extraction should by included in nlu.yml file.
e.g.
nlu:
- intent: inform
  examples: |
    - My account Number is [1234567](acc-No)

square bracket — entity
paranthesis — labels for entities.

- There are three ways entities can be extracted in Rasa.

① Using Pre-built models:
- Duckling for extracting numbers, dates, url email address.

e.g. entity: number
value: 500
component: DucklingHttpExtractor

Spacy: for extracting names, product names, location etc.

e.g.

entity: location
value: Canada
component:
SpacyEntityExtractor

I am looking for a flight to [Canada]{location} that is under [$500]{Number}

② Using Regex:
For entities that match a specific patte
(e.g. phone numbers, postcodes, etc)

My account number is [123456789012]
regex: \d{10,12}

nlu:
- regex: account_number
  examples: |
  - \d{10,12}
- intent: inform
  examples: |
  - My account number is [123456789012](accou
    Numb

③ Using Machine learning.

- For extracting custom entities

Page No.:
Date

Rasa Custom Actions:-

What do we want ~~to~~ virtual assistants to do?
→ Pick up what the user wants from assistant
→ Respond appropriately.
- Send back appropriate message.
- Send an email
- make a calender appointment.
- check info from an API.

```
from typing import Any, Text, Dict, List

from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectionDispatcher

•
Class ActionHelloWorld(Action):
    def name(self) -> Text:
        return "action_name"

    def run(self, dispatcher: CollectionDispatcher,
            tracker: Tracker
            domain: Dict[Text, Any]) ->
            List[Dict[Text, Any]]:
        print("custom code goes here")

        return []
```

CollectionDispatcher → Send messages back to user.

Tracker → To fetch the relevant information
e.g. - Intent
- Entities
- Conversation so far
- Slots

domain → info from domain.yml file