

FAQ CHATBOT

MIDSEM PRESENTATION

Presented by:-
Abhijith Kannan
Harsh Pandey

TABLE OF CONTENTS

01 About Rasa

02 rasa_nlu

03 rasa_core

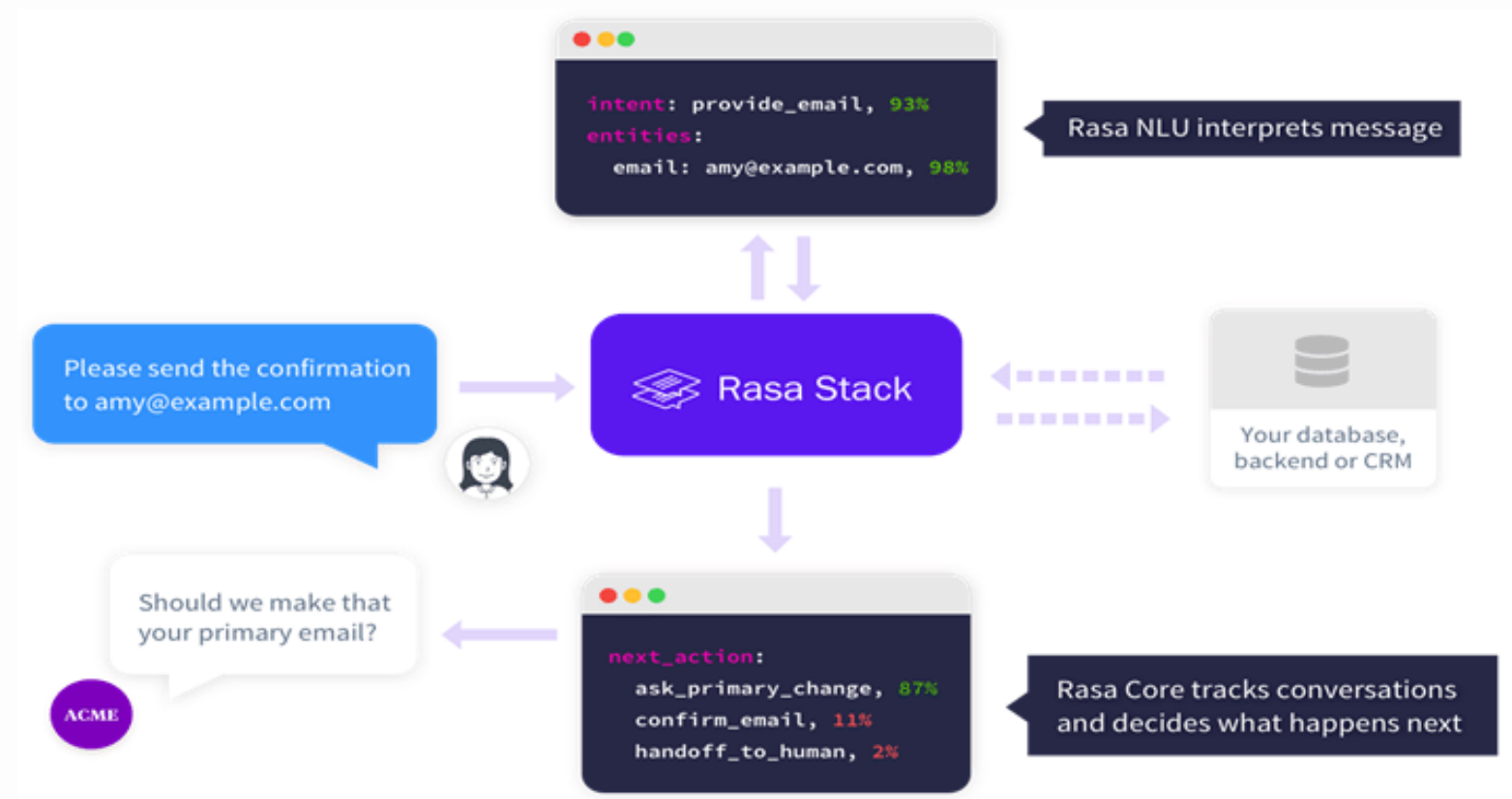
04 prediction policies

05 training data

06 product demonstration

07 question

ABOUT RASA

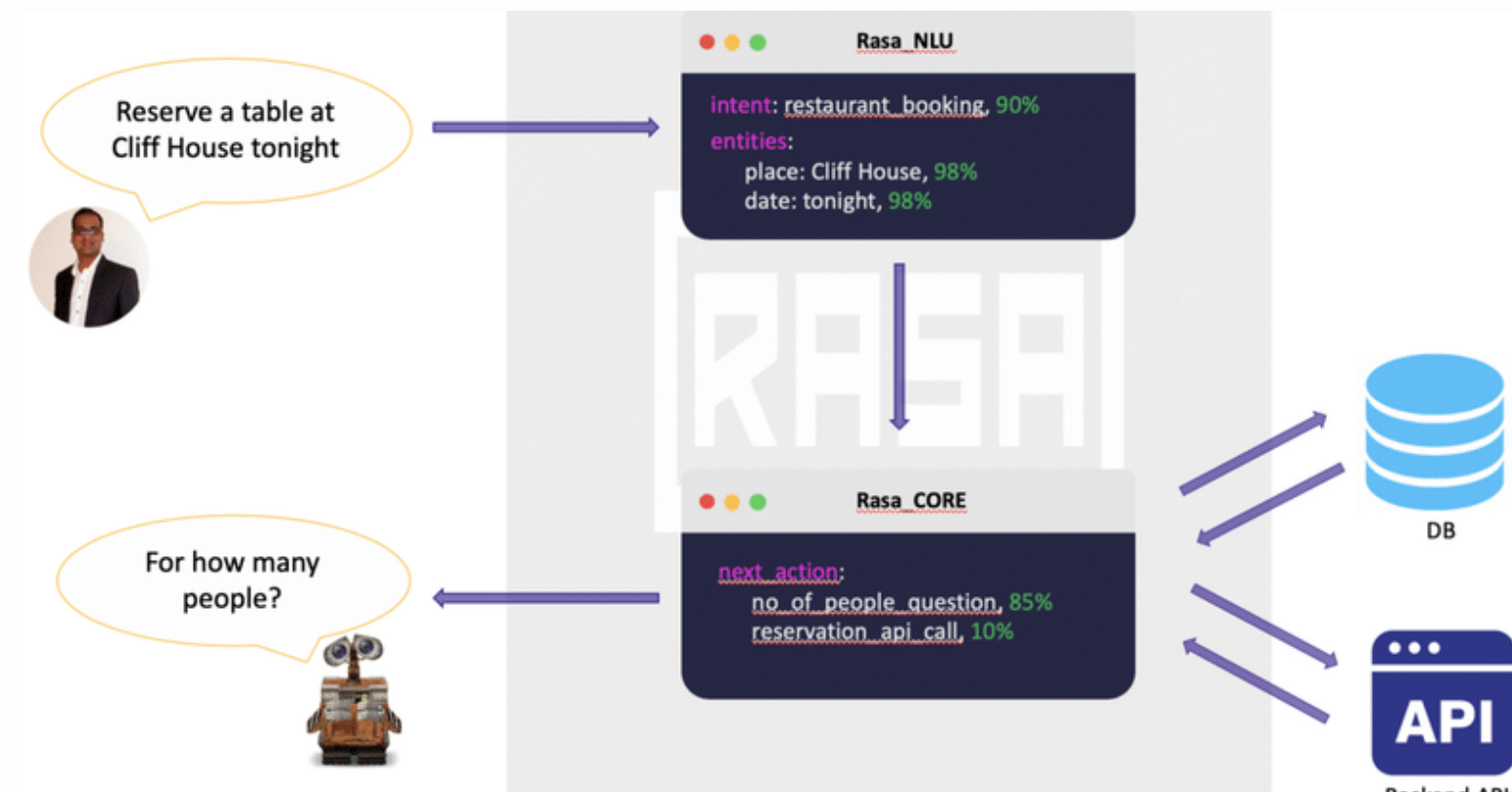


Rasa is a framework for bootstrapping conversational chatbots.

It has two components:

- Rasa NLU
- Rasa Core

RASA NLU

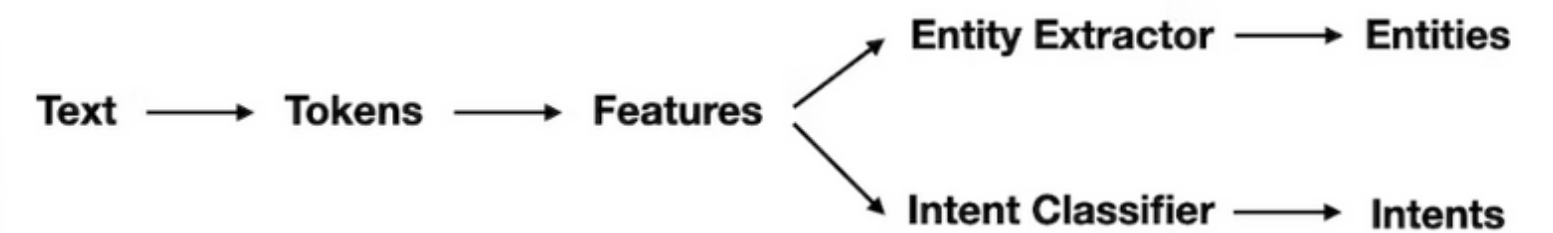


It is a natural language understanding tool for intent classification and entity extraction in chatbots.

The NLU pipeline is defined in the `config.yml` file in Rasa. This file describes all the steps in the pipeline that will be used by Rasa to detect intents and entities.

There are different types of components that you can expect to find in a pipeline. The main ones are:

- Tokenizers
- Featurizers
- Intent Classifiers
- Entity Extractors



CONFIG.YML

```
language: "en_core_web_md"
pipeline:
- name: "SpacyNLP"           # loads the spacy language model
- name: "SpacyTokenizer"     # splits the sentence into tokens
- name: "CRFEntityExtractor"  # uses the pretrained spacy NER model
- name: "SpacyFeaturizer"    # transform the sentence into a vector representation
- name: "SklearnIntentClassifier" # uses the vector representation to classify using SVM
- name: "EntitySynonymMapper" # trains the synonyms
policies:
- name: MemoizationPolicy
- name: TEDPolicy
  max_history: 5
  epochs: 10
```

SpacyTokenizer

- The tokenizer splits each individual word in the utterance into a separate token, and commonly the output of the tokenizer is a list of words.

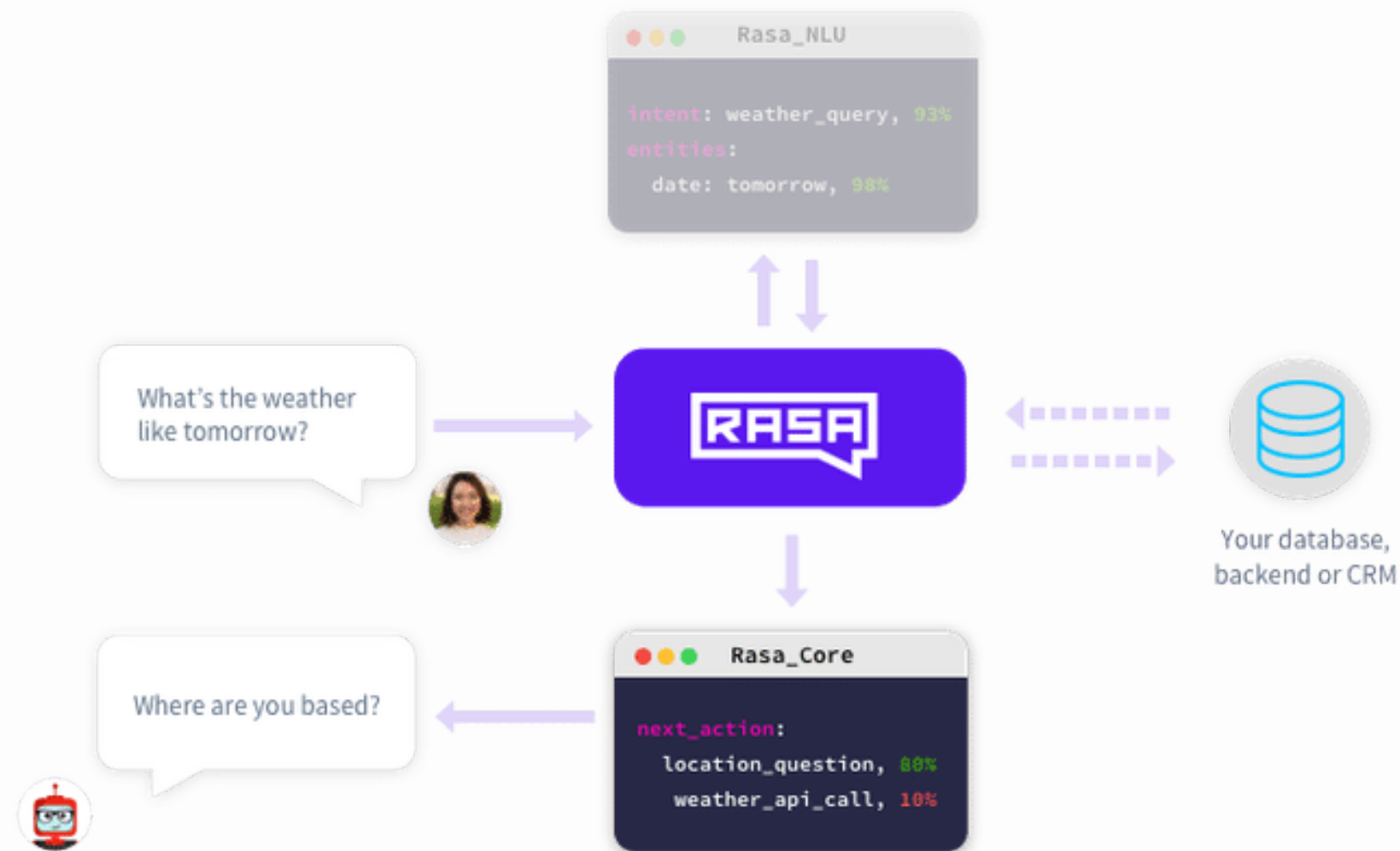
SpacyFeaturizer

- Featurizers generate numeric features for machine learning models.
- There are two types of features:
 - Sparse Features
 - Dense Features

SklearnIntentClassifier

- Once we've generated features for all of the tokens and for the entire sentence, we can pass it to an intent classification model.
- DIET model is used which can handle both intent classification and entity extraction.

RASA CORE



Rasa Core's job is to decide what should happen next during the course of a dialogue

it uses the files,

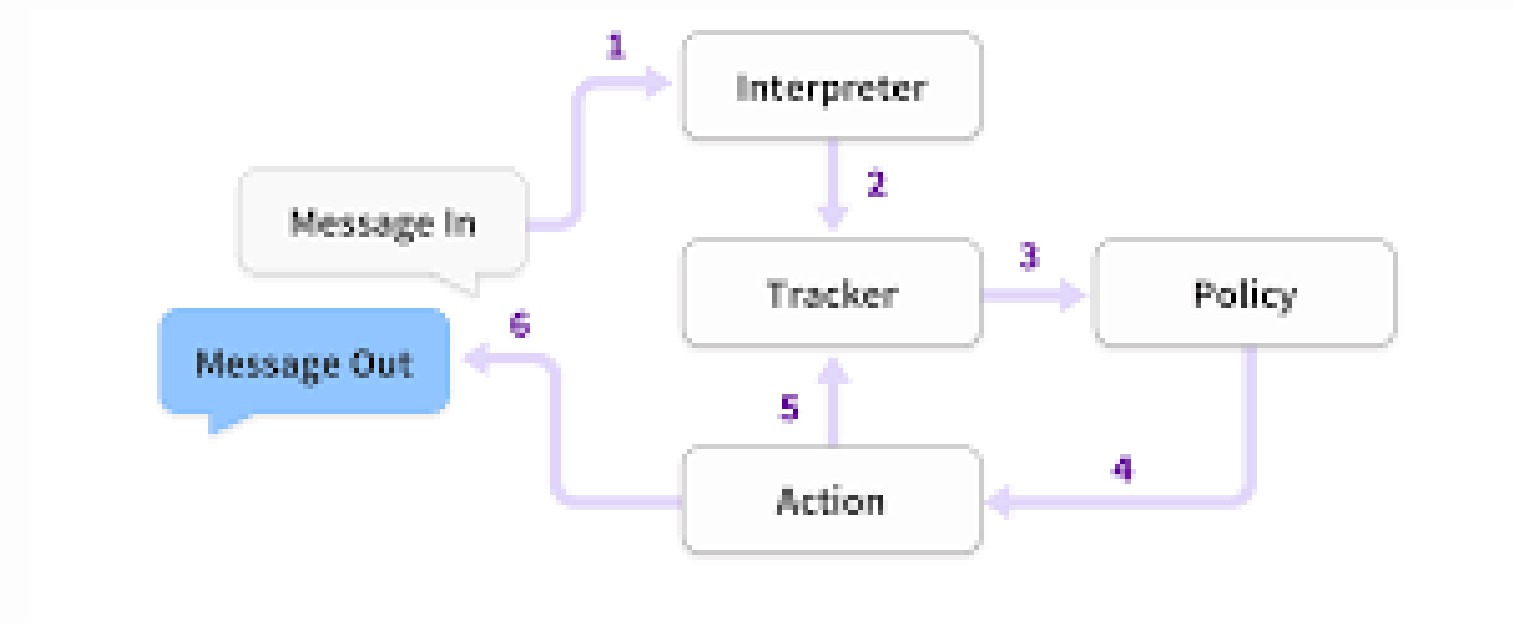
- stories
- domain.yml
- dialogue model

RASA_CORE POLICIES

RASA uses policies to decide which action to take at each step in a conversation. There are machine-learning and rule-based policies that your assistant can use in tandem

- MemoizationPolicy
- TEDPolicy
- RulePolicy

At every turn, each policy defined in your configuration will predict a next action with a certain confidence level. Prediction having max confidence and high priority gets selected



MemoizationPolicy

- remembers the stories from your training data.
- if the current conversation matches the stories in your stories.yml file.
- it will predict the next action from the matching stories of your training data with a confidence of 1.0

TED (transformer embedding dialogue policy)

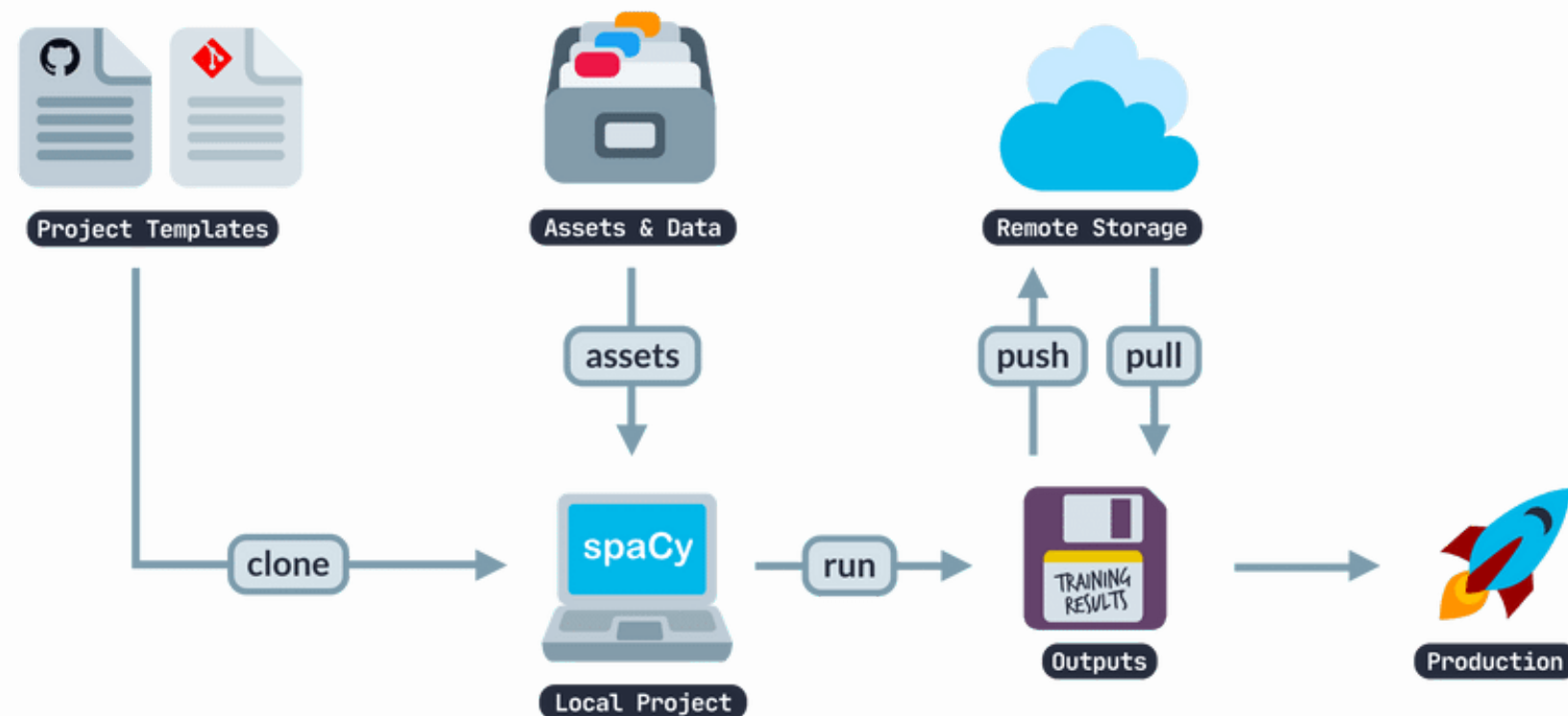
It is a multi-task architecture for next action prediction and entity recognition. The architecture consists of several transformer encoders which are shared for both tasks.

RulePolicy

The RulePolicy is a policy that handles conversation parts that follow a fixed behavior (e.g. business logic).

It makes predictions based on any rules you have in your training data

TRAINING DATA



Our chatbot utilizes a pre-trained model extracted from spaCy library. The language module for the training is "en_core_web_md" which helps in efficiently determining entities and intents using Natural Language Understanding.

The model is fine-tuned using custom training data which is defined in "nlu.md" and "domain.yml" files.

NLU.MD

```
## intent: info_timings
- what are the timings of the temple?
- when can we go to the temple?
- when will the temple be open?
- when should I visit the temple?
- when to visit the temple?
- when to plan visit to temple?
- when does the temple close?
- what is closing time of temple?

## intent: significance
- what is significance?
- what does the temple signify?
- what is special about this temple?
- why was the temple built?
- what does the temple stand for?

## intent: reachability
- how to reach this temple
- what is nearest airport?
- what is nearest railway station?
- how do I reach the temple
- how to plan logistics to reach the temple
- how to travel to this temple
- best route to reach this temple
```

We have fine-tuned our model using the most common intents for our specific use case

- intents : corresponds to respective purpose which user is trying to execute, identified by our model
- followed by the training examples we provided

DOMAIN.YML

```
templates:
  utter_greet:
    - text: "Hey! How are you?"
    - text: "Hey there! How can I help you, today?"
    - text: "Hello! I am here to answer your questions regarding Saraswati Temple."
    - text: "Hola! How can I help?"

  utter_did_that_help:
    - text: "Did that help you?"

  utter_goodbye:
    - text: "Bye"
    - text: "Goodbye!"
    - text: "I hope I solved your queries. Adios!"
    - text: "See ya there!"

  utter_iamabot:
    - text: "I am a bot, powered by Rasa."

  utter_location:
    - text: "The temple is located inside Bits Pilani premises infront of T-lawns."

  utter_reachability:
    - text: "Choose from where you want to visit: Jaipur and Delhi"

  utter_info_timings:
    - text: "The temple is open everyday from morning 8am to evening 7pm (except for special holidays). "
```

Domain.yml file contains the declarations and responses corresponding to every intent defined in the nlu.md file. This mapping is used to return the correct response from the chatbot after it recognizes the intent of the user input.



DEMONSTRATION

basic chatbot link :

<https://colab.research.google.com/drive/1pTuJysl3lXaeKJHxNDULzCoCuE2CE-7b#scrollTo=PyQVtGjWUxdK>

multilingual chatbot link :

https://colab.research.google.com/drive/1UBQ4ljMReBp0jJIKuA962XBRItV_P1Nw#scrollTo=GfWnPccZdtC7

QUESTIONS

