

# UPTYCS INTERNSHIP PROJECT

By Yuvan Pradhan and Zain Muhammed

(4<sup>th</sup> sem students from PES University)

## PROBLEM STATEMENT:

*Design a Q&A system which takes a question as input and returns the intent and entities using a REST based or GRPC based server.*

## RESEARCH:

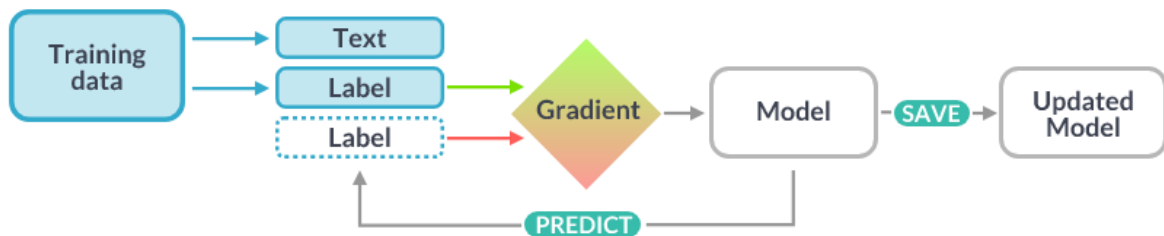
- Studied to get a good understanding of NLP basics.
- Learnt about fundamental concepts (intents, entities, transformers, pipelines, models, tokenizers, featurizers, word embeddings) and tools and libraries used.
- Read through multiple research papers
- In order to solve the problem statement we looked at 3 models out of which we were able to implement 2 of them.

## MODEL 1:

- Joint model for NER (Named Entity Recognition) and IC (Intent Classification) using BERT

- BERT (Bidirectional Encoder Representations from Transformers)
- A specific, large transformer masked language model based on deep learning
- Difficult to implement due to a lack of data
- Research Paper: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9599152>

## MODEL 2:



- Custom NER model using SpaCy
- Data annotation
- Load pre-trained SpaCy model
- Converted annotated training data to SPACY file
- Download config file
- Train model with converted dataset
- Load the model and evaluate with inputs
- Could not do both NER and IC on the same model so we tried RASA NLU

```
nlp1 = spacy.load(r".\output\model-best") #load the best model
doc = nlp1("Who can access route table?") # input sample sentence

spacy.displacy.render(doc, style="ent", jupyter=True)
```

✓ 7.8s

Who WHO can access ACCESS route table ROUTE\_TABLE ?

### MODEL 3:

- NER and IC using RASA NLU
- Set up default RASA environment
- Modify domain.yml by adding a list of intents and entities to it
- Modify nlu.yml by adding examples of intents with various statements/regex
- Highlight entities in the example statements in nlu.yml
- Train NLU model
- Run RASA server on a port of your choosing to send HTTP requests to it (we used Postman to send these requests and receive responses)

POST localhost:5005/model/parse

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL Text

```

1  {
2    "text": "Who can access account 123456"
3  }

```

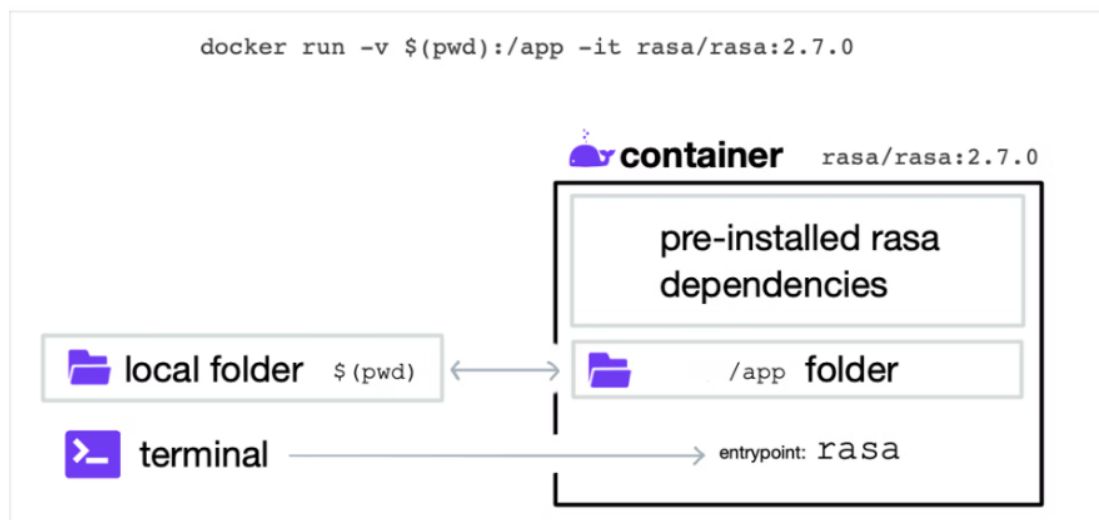
Body Cookies Headers (6) Test Results 200 OK 29 ms 1.55 KB

Pretty Raw Preview Visualize JSON

```

1  {
2    "text": "Who can access account 123456",
3    "intent": {
4      "id": 8880379122523121179,
5      "name": "access",
6      "confidence": 0.9943814873695374
7    },
8    "entities": [
9      {
10     "entity": "account_id",
11     "start": 23,
12     "end": 29,
13     "confidence_entity": 0.6289570331573486,
14     "value": "123456",
15     "extractor": "DIETClassifier"
16   }

```



To run on Docker,

- Load premade RASA container on Docker
- Run RASA server using Docker

```

(myenv) zain@ZAINS-PC:/mnt/c/Users/zainr/Documents/rasa_nlu_project$ docker run -it -p 8080:8080 -v $(pwd):/app rasa/rasa:2.7.0
a:2.8.0 run --enable-api --port 8080
2022-07-03 07:17:43 INFO root - Starting Rasa server on http://localhost:8080
2022-07-03 07:17:43 INFO rasa.model - Loading model models/nlu-20220702-104827.tar.gz...
2022-07-03 07:18:06 INFO root - Rasa server is up and running.

```

