Chatbot Architecture

The chatbot is made to be provided as a pluggable service, and configurable through configuration files, in addition to modifying code.

Concepts

Intent : This is a particular skill of a chatbot. It could be booking a flight, or searching a store in a location etc.
Each intent has a few **parameters**, which are basically information required to be elicited by the user for the chatbot to complete the particular intent. Example : Book name, city name, registration number etc.
Actions : Each intent also had an action, that is basically a function call after all the parameters have been fulfilled.

Configuration Files:

Entities Folder:

Here each file contains example entities.
Example -

```
anirudhdahiya:~/karma/nlp/chatbotPOC/entities$ ls
author.dat      location.dat    storenames.dat  title.dat
anirudhdahiya:~/karma/nlp/chatbotPOC/entities$ cat storenames.dat
Dominos
KFC
Pizza Hut
Subway
Airtel
Vodafone
Decathlon
anirudhdahiya:~/karma/nlp/chatbotPOC/entities$
```

Intents Folder:

Here each file contains example utterances for each intent. Note the placeholders starting with $ for entity masking (as done by attribute_getter)
Example -

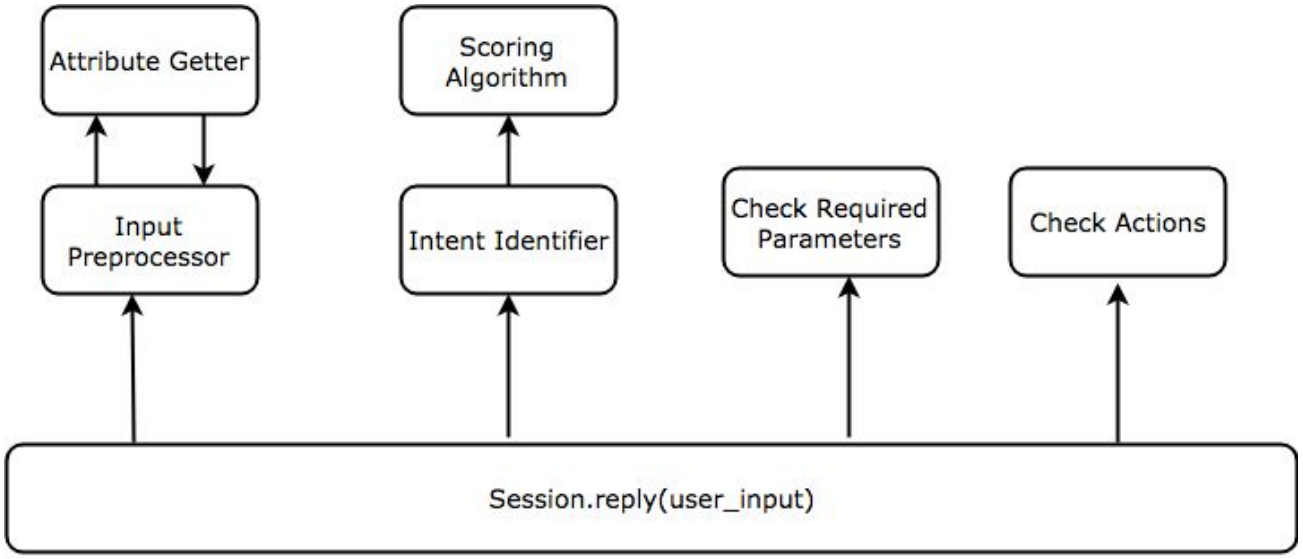
```
anirudhdahiya:~/karma/nlp/chatbotPOC/intents$ ls
book.dat    search.dat
anirudhdahiya:~/karma/nlp/chatbotPOC/intents$ cat book.dat
I want to order a book
order book
I want to order $author book
I want to order $title by $author
I want to order $title
anirudhdahiya:~/karma/nlp/chatbotPOC/intents$
```

params/newparams.cfg :

This file contains the json which holds the parameters for each intent, and its corresponding values, such as placeholder, required etc.

```
anirudhdahiya:~/karma/nlp/chatbotPOC/params$ ls
newparams.cfg
anirudhdahiya:~/karma/nlp/chatbotPOC/params$ cat newparams.cfg
{
        "OrderBook": {
                "intentname": "OrderBook",
                "Parameters": [{
                                "name": "title",
                                "placeholder": "$title",
                                "required": "True",
                                "prompts": [
                                        "What's the name of the book?"
                                ],
                                "context": "FlightBook_dialog_From"
                        },
                        {
                                "name": "author",
                                "placeholder": "$author",
                                "required": "True",
                                "prompts": [
                                        "What's the author name?",
                                        "Who wrote this book?"
                                ],
                                "context": "FlightBook_dialog_To"
                        },
                        {
                                "name": "RegNo",
                                "placeholder": "$RegNo",
                                "required": "True",
                                "prompts": [
                                        "What's your registration number?",
                                        "Please type your registration number"
                                ],
                                "context": "GetRegNo"
                        }
                ],
                "actions": "BookIssue"
        },
        "StoreSearch": {
                "intentname": "StoreSearch",
                "Parameters": [{
                                "name": "location",
                                "placeholder": "$location",
                                "required": "True",
                                "prompts": [
                                        "In which city you want to look up the store in?"
                                ],
                                "context": "FlightSearch_dialog_From"
                        },
                        {
                                "name": "storenames",
                                "placeholder": "$storenames",
                                "required": "True",
                                "prompts": [
                                        "What is the store name?",
                                        "Please mention the store name."
                                ],
                                "context": "FlightSearch_dialog_To"
                        }
                ],
                "actions": "SearchStore"
        }
}
```

Conversation Flow :



First uinput is passed to input preprocessor, where things like spell check can happen. From there attribute getter is called to fetch any entities is user input, and mask user_input with entity placeholders in place of entities.

Intent identification based on masked user input. Scoring algorithm generates score for intent identification based in stored sample intent utterances.
This can be followed by logical decision making based on contexts at intent identifier to move to a new intent or stay at the same

Goes through required parameters of an intent and generates a reply for user to provide that parameter.

Reply

When all parameters satisfied, bot does the intent action (as specified in the configuration file), and generates a reply.

Reply