A Seminar Report
On

Dialogue Intent Identification, Entity Extraction and Slot Filling For Closed Domain Conversational Agents

Submitted By:

Sweety S Sindhav
Enrolment Number: 180280723013
Semester: II,

M.E. (I.T.)

Guided By:

Prof. Pradip R Patel
Prof. Manoj B Patel
Prof. Ankit C Patel

A Seminar Report Submitted
Gujarat Technological University in Partial
Fulfilment of the Requirements for
The Degree of Master of Engineering
In Information Technology

Year: 2019

L. D. College of Engineering
Opp. Gujarat University,

Navrangpura, Ahmedabad - 380015

GOVERNMENT OF GUJARAT

L. D. COLLEGE OF ENGINEERING

AHMEDABAD – 380015

લા.દ.ઈજનેરી મહાવિધાલય,અમદાવાદ-૩૮૦૦૧૫

Phone :- 079-26306752/0735 **Gram : ENCOPRI**
website: http://www.ldceahd.org **email: ldce-abad-dte@gujarat.gov.in**

Date:_/__/___

CERTIFICATE

This is to certify that the work presented in the Mini Project with seminar (3720001) entitled **“Dialogue Intent Identification, Entity Extraction and Slot Filling for Closed Domain Conversational Agents”** has been carried out by **Sweety Sindhav** Enrollment No.:**180280723013** at Department of Information and Technology of L.D.College of Engineering, Ahmedabad under my guidance as a partial fulfilment of requirements to Award ME (Information Technology) By Gujarat Technological University, Ahmedabad. This Seminar Work has been evaluated and found satisfactory.

Internal Guide team:

1) **Prof. Pradip R Patel**
2) **Prof. Manoj B Patel**
3) **Prof. Ankit C Patel**

Head of Department:

Dr. H.M.Diwanji

ABSRTACT

A conversational agent (CA) or a dialogue system is a computer system intended to converse with a human, with a coherent structure. People interact with systems more and more through voice assistants and chatbots. This research will investigate how advancements in Artificial Intelligence and Machine Learning technology are being used to improve many services.

Dialogue systems are of two types - open domain and closed domain system. Open domain conversational systems do not have any well-defined goal. Conversation in these systems can be on infinite number of topics. On the contrary, closed domain conversational agents try to achieve a very specific goal or objective.

Intent identification is an integral part of every conversational system. Dialogue intent identification is the process of identifying the underlying goal of a particular dialogue. A response to any query or dialogue can be generated only when the intent of the query is correctly identified. Thus intent identification is an indispensable component of every conversational system.

Entity Extraction of the query will let us know about which entity the user is talking and thus help us serve better. Entity Extraction will let us know if the entity type and its value which in turn is used to appropriate responses. Thus Entity Extraction is the foremost step for dialogue generation.

TABLE OF CONTENT

Sr.	**Chapter**	Page
	Abstract	
1	Introduction	1
	1.1 Introduction	1
	1.2 Chatbot History	1
2	Types of Chatbot	2
3	Chatbot developing approach	2
4	Chatbots in Industry	4
5	Literature Review	5
	5.1 Intent Detection and Slots Prompt in a ClosedDomain	5
	5.2 Comparison of Multinomial Naive Bayes Algorithm & Logistic Regression for Intent Classification In Chatbot	5
	5.3Intent Identification for knowledge based Question Answering	6
6	Theoratical Basis	7
	6.1 Natural Language Processing	7
	6.2 Rasa Stack	8
7	Implementation	9
	7.1 Programming Environment	9
	7.2 Datasets	10
	7.3 Training the Model	11
	7.4 Testing the model	12
8	Conclusion	14
10	References	15

Introduction

1.1 Introduction

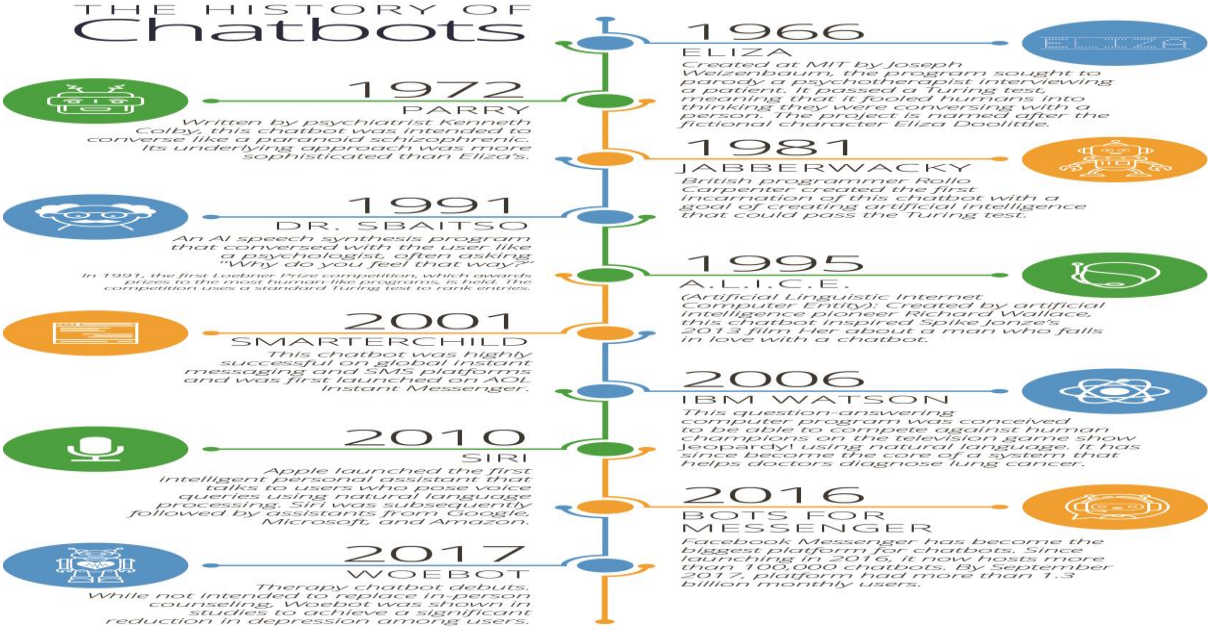
“Digitalisation, the surge of mobile and internet connected devices has revolutionised the way people interact with one another and communicate with businesses” (Eeuwen, M.V. (2017)). Millennial are accepting and supporting new technology into the routine of their everyday life, this is becoming more prevalent as technology companies are streamlining Artificial Intelligence (AI) into the products they offer, such as; Google Assistant, Google Home and Amazon Alexa. The new and upcoming generation are expected to be critical and game changing customers for businesses. “They demand effortless experiences, answers within seconds, not minutes and more intelligent self-service options”

A chatbot is a software tool that utilises natural language processing (NLP) for human machine interaction (HMI) and Machine Learning (ML). “The complexity of a chatbot is directionally proportional to the scope of the domain”. An open domain requires a larger knowledge base, whereas, a closed domain has a more specific knowledge base that was developed to achieve a specific goal.

Intent identification is the process of deducing the goal or meaning of the sentence. Intent identification is performed using various classification algorithms. Performance of dialogue systems is vastly dependent on the accuracy of these intent identification methods and algorithms.

1.2 Chatbot History

Chatbot technology initially began in the 1960s to determine whether a chatbot could be portrayed as a human. Throughout the 1980s there was an elevated amount research carried out on natural language interfaces which lead to the development of sophisticated chatbot architectures such as A.L.I.C.E. This is a chatbot you can create through interaction as it will learn from previous interactions to create its knowledge base. Its knowledge is saved in AIML (Artificial Intelligent Mark-up Language) files which evolved from the Extensible Mark-up Language (XML)



1966 ELIZA

Created at MIT by Joseph Weizenbaum, the program sought to parody a psychotherapist interviewing a patient. It passed a Turing test, meaning that it fooled humans into thinking they were conversing with a person. The project is named after the fictional character Eliza Doolittle.

1972 PARRY

Written by psychiatrist Kenneth Colby, this chatbot was intended to converse like a paranoid schizophrenic. Its underlying approach was more sophisticated than Eliza's.

1981 JABBERWACKY

British programmer Rollo Carpenter created the first incarnation of this chatbot with a goal of creating artificial intelligence that could pass the Turing test.

1991 DR. SBAITSO

An AI speech synthesis program that conversed with the user like a psychologist, often asking "Why do you feel that way?"

In 1991, the first Loebner Prize competition, which awards prizes to the most human-like programs, is held. The competition uses a standard Turing test to rank entries.

1995 A.L.I.C.E.

(Artificial Linguistic Internet Computer Entity): Created by artificial intelligence pioneer Richard Wallace, this chatbot inspired Spike Jonze's 2013 film Her about a man who falls in love with a chatbot.

2001 SMARTERCHILD

This chatbot was highly successful on global instant messaging and SMS platforms and was first launched on AOL Instant Messenger.

2006 IBM WATSON

This question-answering computer program was conceived to be able to compete against human champions on the television game show Jeopardy! using natural language. It has since become the core of a system that helps doctors diagnose lung cancer.

2010 SIRI

Apple launched the first intelligent personal assistant that talks to users who pose voice queries using natural language processing. Siri was subsequently followed by assistants from Google, Microsoft, and Amazon.

2016 BOTS FOR MESSENGER

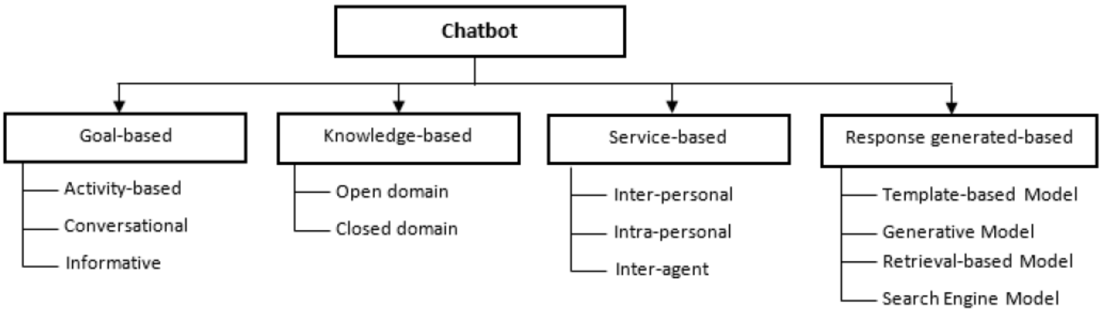
Facebook Messenger has become the biggest platform for chatbots. Since launching in 2016, it now hosts more than 100,000 chatbots. By September 2017, platform had more than 1.3 billion monthly users.

2017 WOEBOT

Therapy chatbot debuts. While not intended to replace in-person counseling, Woebot was shown in studies to achieve a significant reduction in depression among users.

2. Types of chatbot

Dialogue systems are of two types - open domain and closed domain system. Open domain conversational systems do not have any well-defined goal. Conversation in these systems can be on infinite number of topics. On the contrary, closed domain conversational agents try to achieve a very specific goal or objective.



3. Chatbot developing Approach

Chatbots are developed using two approaches; a **rule based approach** where chatbots operate by moving through branches of a tree diagram of an expert system. The second approach involves **advanced artificial intelligence and machine learning**, thus the chatbot can learn from the conversations, generating appropriate responses to continuously improve over time (Watson, A. 2017).

3.1 Problem with rule based System

When computer start understanding English,its all about NLP {Natural Language Processing} . If you do not use NLP in chatbot , It will not be able to recognize the intent . Suppose You Train your machine with a question –

Question- What is chatbot ?

Answer – Its AI based human conversation simulation **.**

Now If you ask **"Tell me about chatbot "** , It will not recognize it with the same question earlier asked . Now think How many type of similar question you need to train your Machine to achieve a single Intent . There could be thousands variation of a single intent .Therefore intent identification using machine learning plays crucial role for building conversational agent.

3.2 Chatbot and Machine learning

Machine learning chatbots works using artificial intelligence. User need not to be more specific while talking with a bot because it can understand the natural language, not only commands. This kind of bots get continuously better or smarter as it learns from past conversations it had with people. Here is a simple example which illustrate how they work. The following is a conversation between a human and a chatbot:

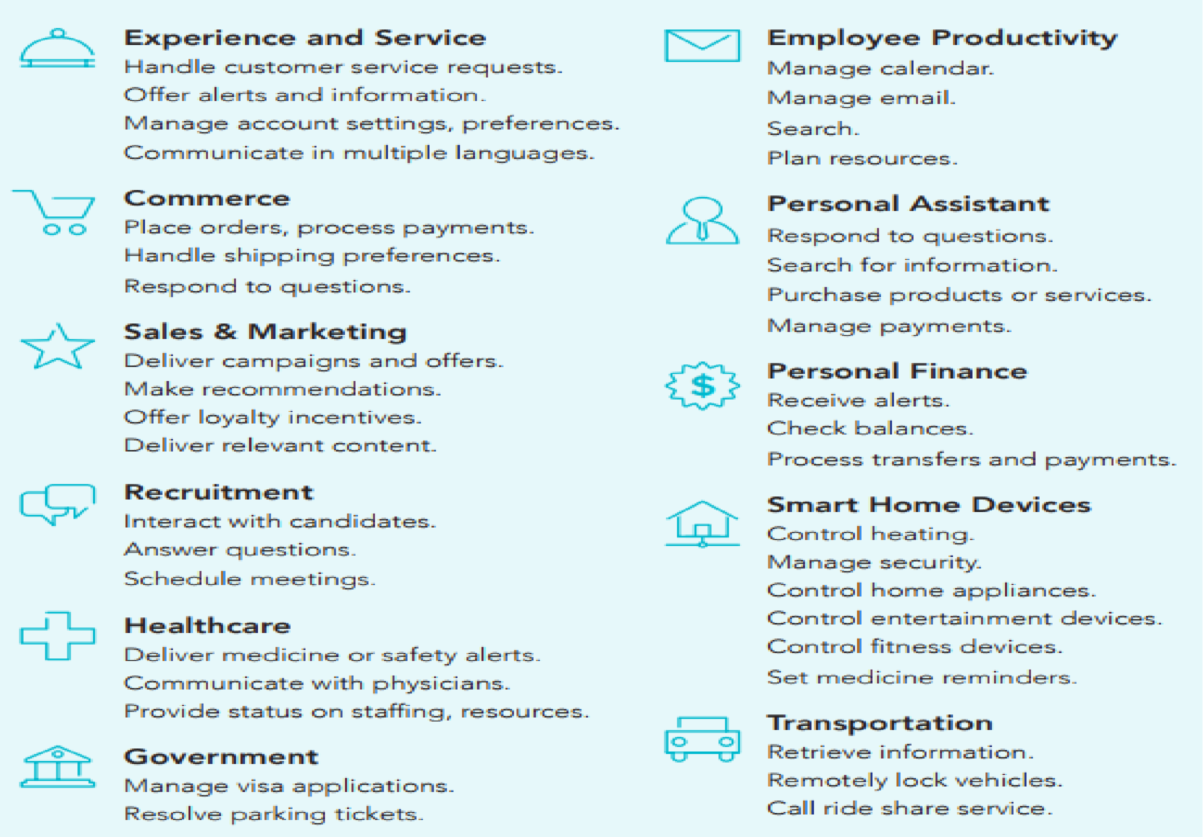
Human: “I need a flight from San Jose to New York.”
Bot:“Sure! When would you like to travel?” Human: “From Dec 20, 2016 to Jan 28, 2017.”
Bot:“Great! Looking for flights.”

For the Rule Based Conversational Agent it can not determine the missing entity. In the above example by using Machine Learning, it helped the model to recognize the missing entity and ask user about the same. This task is known as slot-filling.

Natural language processing (NLP) is the processing of human -- and not computer – language by a computer program. One of the older and best known examples of NLP is spam detection, which looks at the subject line and the text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.

4 Chatbots in Industry

Most businesses and organisations are understanding the potential benefits of machine learning and artificial intelligence to have a positive change on how they perform business. Artificial intelligence has progressed to allow the development of more sophisticated chatbots. Organisations are focusing on specific areas of user engagement that take up a lot of time but can be replaced through the use of a chatbot. Chatbots can understand what the customer needs from a single text instead of the customer having to follow a process of multiple steps.



Experience and Service
Handle customer service requests.
Offer alerts and information.
Manage account settings, preferences.
Communicate in multiple languages.

Commerce
Place orders, process payments.
Handle shipping preferences.
Respond to questions.

Sales & Marketing
Deliver campaigns and offers.
Make recommendations.
Offer loyalty incentives.
Deliver relevant content.

Recruitment
Interact with candidates.
Answer questions.
Schedule meetings.

Healthcare
Deliver medicine or safety alerts.
Communicate with physicians.
Provide status on staffing, resources.

Government
Manage visa applications.
Resolve parking tickets.

Employee Productivity
Manage calendar.
Manage email.
Search.
Plan resources.

Personal Assistant
Respond to questions.
Search for information.
Purchase products or services.
Manage payments.

Personal Finance
Receive alerts.
Check balances.
Process transfers and payments.

Smart Home Devices
Control heating.
Manage security.
Control home appliances.
Control entertainment devices.
Control fitness devices.
Set medicine reminders.

Transportation
Retrieve information.
Remotely lock vehicles.
Call ride share service.

5. Literature Review

In this chapter, we have studied some of the research papers to boost the seminar topic. There is a number of researches study have been accomplished and many studies are going on in this field. In this work, some of the relevant research papers have been studied and have been written the observations best of our knowledge from those papers.

5.1 Intent Detection and Slots Prompt in a Closed-Domain Chatbot:

Predicting intent and slots of a query for a chatbot that answers career related queries. Used a multi-staged approach where both the processes (intent-classification and slot-tagging) inform each other's decision-making in different stages.

Method:
Classify intent of a user-query into categories and subcategories using RNNs, and find entities using our Named Entity Taggers in multiple stages. The above two tasks are accomplished in a parallel fashion wherein both the tasks benefit from the other's intermediate results. Using Stanford CoreNLP to train the taggers using different vocabularies. Collected user queries over a period and used some publicly available non - native datasets for training taggers.

Research gap:

Extend this work by detecting synonymous phrases in the user queries to increase the coverage of variations in queries that can be handled. Further plan to include more categories and subcategories of problems and to test algorithm on other public datasets.

5.2 Comparison of Multinomial Naive Bayes Algorithm And Logistic Regression for Intent Classification In Chatbot

The researcher proposes a classification method to identify intent rather than user input or called intent classification on the chatbot system; the researcher also wants to know the level of accuracy, precision, and recall on the evaluation results of both methods. The classification method applied in this research is the Naive Bayes method and compared with the Logistic Regression method to determine the class intention. The evaluation results show the level of accuracy precision and recall in the Logistic Regression model is higher than the Naive Bayes model.

Method:
Retrieve the data in the study was reading the training data that had been labeled in the class intent that had been done by the researcher, and the researcher made the class as follows greet, report, info, point, trade-point, and thanks. In the training data, some data is taken to be used as a data test to test the level of accuracy and performance of each model at the evaluation stage. After retrieving the data the researcher processes the TF-IDF.
After being transformed using TF-IDF then doing the data sharing stage into training data and test data.In the training data section, the model was made by applying the Naive Bayes method and Logistic Regression to detection of classification.

After getting the classifier model created using training data, the next step is to evaluate the data using test data taken by 20% of the training data. At the time of retrieving the data, by predicting test data on training data in both methods, it produces an evaluation in the form of accuracy, precision, and recall value by using a confusion matrix.

5.3 Intent Identification for knowledge based Question Answering

This research proposed an intent identification method to wipe off irrelevant words to decrease the semantic influence. By locating, expanding and disambiguating the subject and its attributes of questions, it reduces the time of KBQA and also reduce amount of data processing and search space. By incorporating Convolutional Neural Network to model questions and answer candidates, the top ranking candidates can be easily identified as answers.

Method:
Firstly identified both subject and its attribute from the question so as to reduce the semantic interference caused by semantic irrelevant words.

Then apply an advanced entity linking system to generate the candidate subjects with a candidate subject directory, thus pre screening related triples in the KB as candidate answers.

6.Theoratical Basis

6.1 Natural Language Processing

Natural Language Processing (NLP) is the study of letting computers understand human languages. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers don't recognize the words and don't understand the grammars. NLP can be regard as a "translator", who will translate human languages to computer understandable information. Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. A single replace of one character or even a space can have significant difference. However, the emergence of NLP is changing the way of interacting. Apple Siri and Microsoft Cortana have made it possible to give command in everyday languages and is changing the way of interacting.

"Natural Language Processing (NLP) is a theory motivated range of computational techniques, for the automatic analysis and representation of human language" (Jurafsky , D and Martin, J.H. 2017). Natural Language Processing technology has made great advancements in machine learning based systems to be able to extract meaning from natural language utterances also known as sentiment analysis (Cambria and White, 2014).

There are many techniques used in NLP for the analysis of natural language
Entity recognition is a technique which recognizes entities in text. The most common entities include; nouns, organisations, people and places.

Named Entity Recognition (NER) is the task of finding every instance of a named entity in text and label its type in order to classify it correctly."Entities are domain specific information extracted from the utterance that maps the natural language phrases to their canonical phrases in order to understand the intent. They help in identifying the parameters which are required to take specific action"

Establishing the **context** of the of the users message is a vital feature that allows the chatbot to deal with situations that it may not be able to carry out a specific action for. This is due to the user input being very vague or may have an alternative meaning. The context is the capacity of a chatbot to sustain its state, also referred to as the number of user supplied input (utterances) when the context is extracted and the appropriate intent is paired to conduct the desired action for the user.

Intents are the core backbone of conversational interfaces. The intents represent what the customer is wanting to achieve such as 'show me my balance'. The text sent by the user in natural language is analysed to identify the corresponding intent of the text. This requires matching a specific user supplied phrase with an appropriate action to be executed by the system. The chat bot would then return appropriate dialog in order for the user to achieve their goals.

Actions are the processes or steps executed by the system when the intent of a message is identified, they contain parameters which categorise and define its properties.
Sentiment analysis incorporates multiple natural language processing techniques in order to extract meaning and polarity from text. Polarity detection classifies text to be either positive or negative and measures the intensity of the overall polarity detected. Sentiment analysis achieves a more in depth understanding of the contextual role of each concept within a given piece of text.

Part of Speech (POS) Tagging involves assigning a label to each word in the user input with its part of speech (e.g. noun, verb, adjective, etc.). The labels or tags can be rule-based (a manually developed set of rules is defined to establish part of speech for ambiguous words provided in the conversational context). The labels can also be developed utilising advanced models that are trained using input labelled with the appropriate POS. This is additionally used in response generation in order to outline the POS object type of the expected response made by the chat bot.

6.2 Rasa Stack

The proposed system applies Machine Learning and deep learning algorithm using Rasa Stack, particularly Rasa_nlu and Rasa_core. This in turn help the conversational agent to generate output.

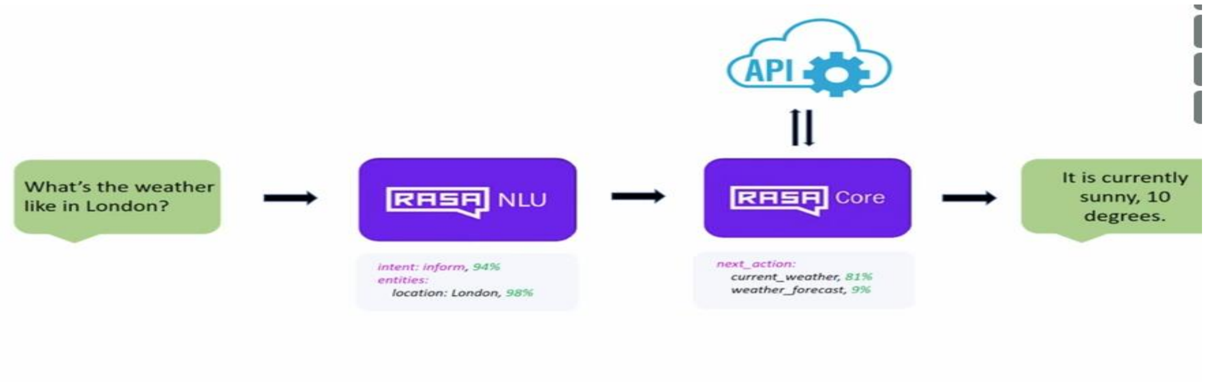
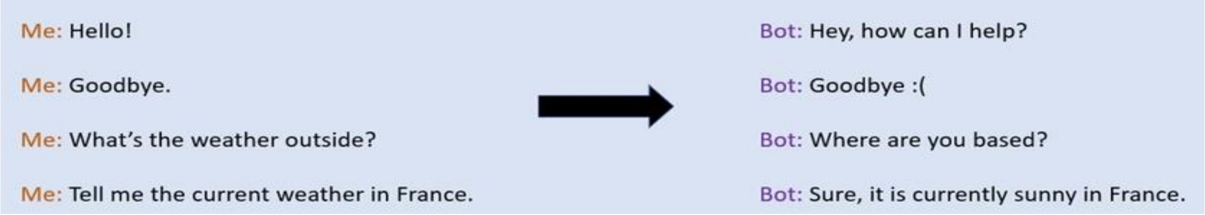


Fig : Rasa Stack : Rasa_nlu and Rasa_core

The model in this project uses machine learning to identify intent and extract entity and its value. The model is smart enough to recognize if the entity is missing and in turn asks the user to provide the entity.



Me: Hello!

Me: Goodbye.

Me: What's the weather outside?

Me: Tell me the current weather in France.

Bot: Hey, how can I help?

Bot: Goodbye :(

Bot: Where are you based?

Bot: Sure, it is currently sunny in France.

Fig: ChatBot using Machine Learning

7. Implementation

7.1 Programming Environment

Python programming language is used to implement Intent Identification and Entity Extraction in the proposed model as it supports handful of libraries making implementation of machine learning easier for developer. Also needs to setup a virtual environment for this project as the Packages version may cause inconsistency with the base environment.

Libraries used:

- **Rasa Stack**

Machine Learning libraries to create contextual AI assistants and chatbots that go beyond simple questions. It is a Open source natural language processing and dialogue management.

  - **Rasa_nlu** : Used for natural Language Understanding. It provides libraries for intent classification and Entity Extraction
  - **Rasa_core**: Used for Natural Lnaguage Generation. It provides libraries for dialogue generation.

- **Tensorflow**

  TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks

- **Keras**

  Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible

- **Scikit-learn**:

  Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language.

  It featuresvarious classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

- **SpaCy**

  spaCy is an open-source software library for advanced Natural Language Processing, written in the programming languages Python and Cython

- **sklearn-crfsuite**

  sklearn-crfsuite is thin a CRFsuite (python-crfsuite) wrapper which provides scikit-learn-compatible sklearn_crfsuite

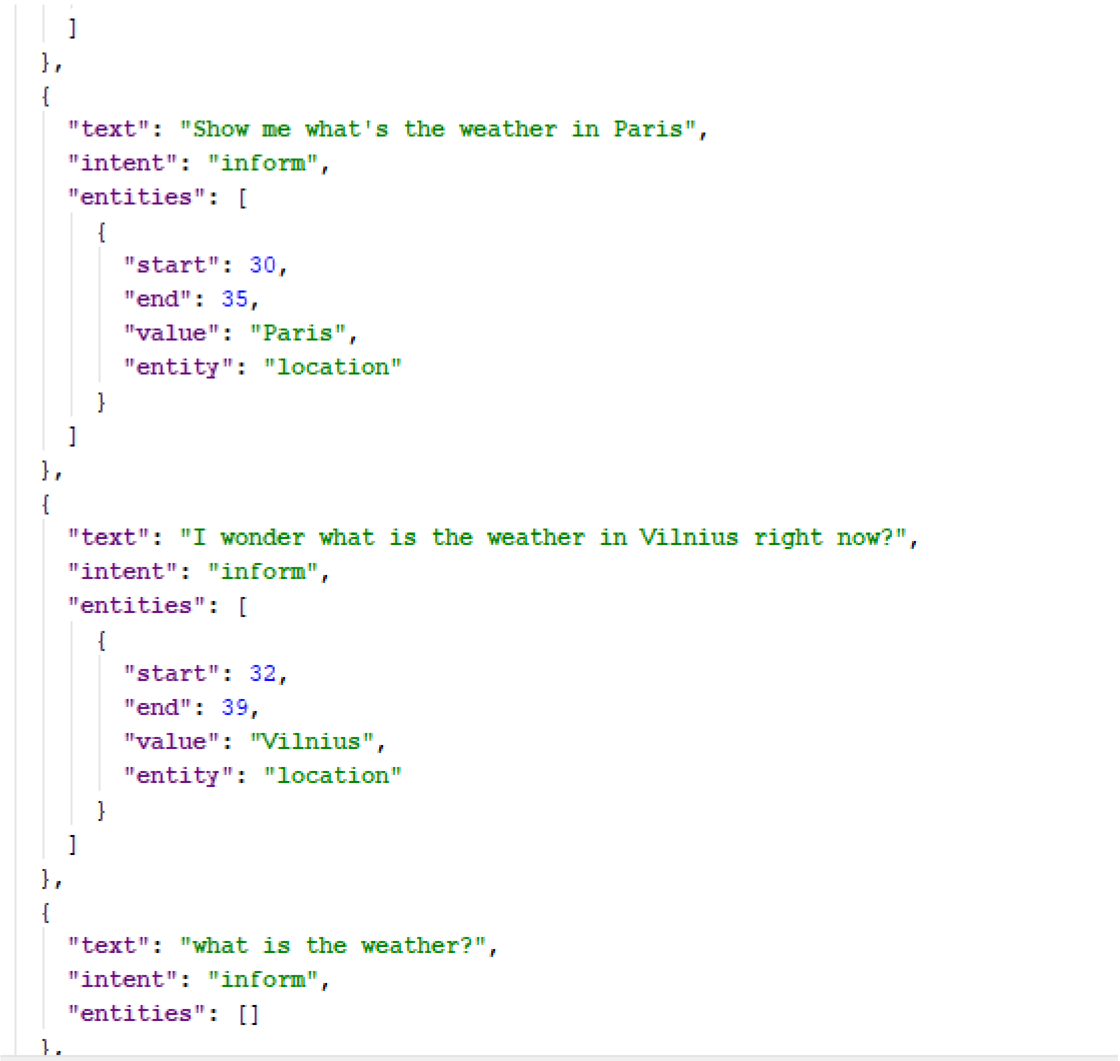
- **git+https://github.com/apixu/apixu-python.git**

  Apixu offers free Weather API (JSON and XML) and Geo API as a simple, affordable and fully managed feed for programmers, developers, businesses

Other basic machine learning libraries such as numpy, scrapy, matplotlib etc

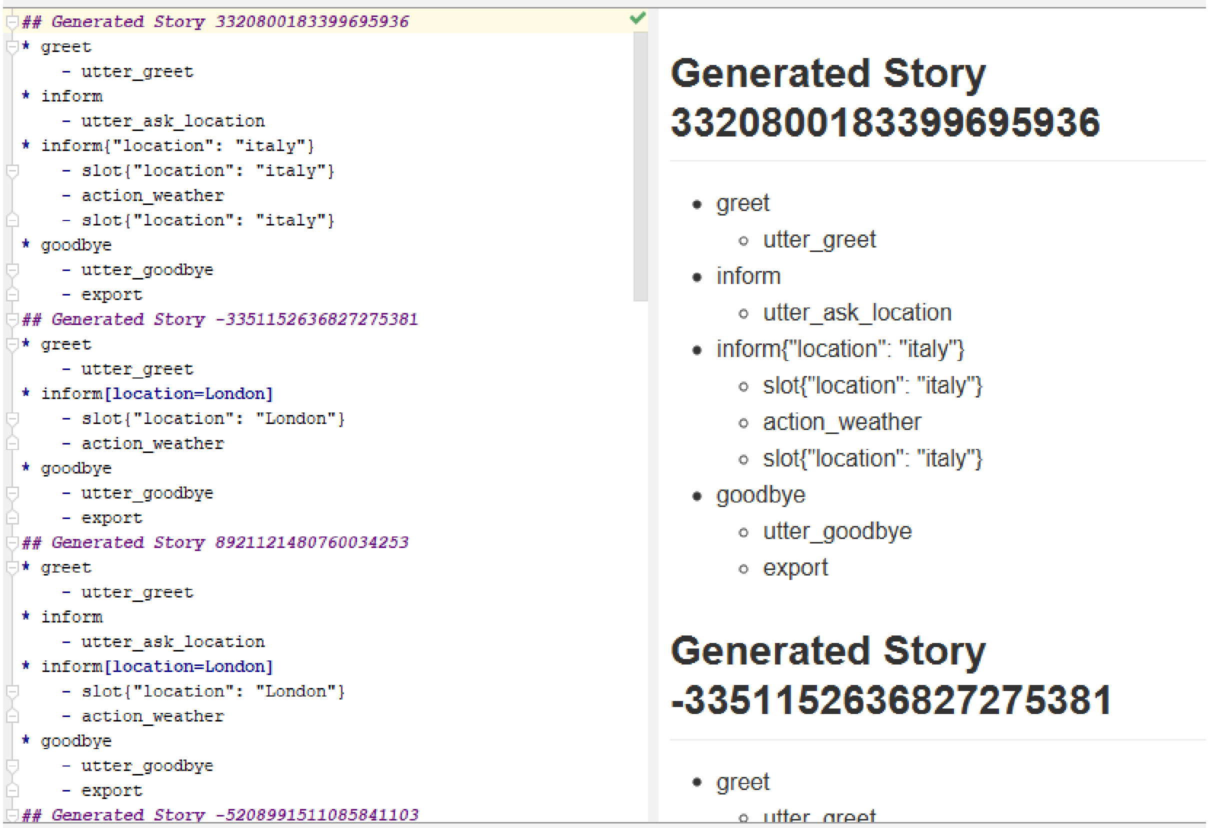
7.2 Datasets

A Json and markdown file is prepared which consist of intent and entity example to train the model. Also the data is created on the go by generating stories as we chat with the model.



```
    ]
  },
  {
    "text": "Show me what's the weather in Paris",
    "intent": "inform",
    "entities": [
      {
        "start": 30,
        "end": 35,
        "value": "Paris",
        "entity": "location"
      }
    ]
  },
  {
    "text": "I wonder what is the weather in Vilnius right now?",
    "intent": "inform",
    "entities": [
      {
        "start": 32,
        "end": 39,
        "value": "Vilnius",
        "entity": "location"
      }
    ]
  },
  {
    "text": "what is the weather?",
    "intent": "inform",
    "entities": []
  },
```

Fig: Dataset for training the model



```
## Generated Story 3320800183399695936
* greet
    - utter_greet
* inform
    - utter_ask_location
* inform{"location": "italy"}
    - slot{"location": "italy"}
    - action_weather
    - slot{"location": "italy"}
* goodbye
    - utter_goodbye
    - export
## Generated Story -3351152636827275381
* greet
    - utter_greet
* inform[location=London]
    - slot{"location": "London"}
    - action_weather
* goodbye
    - utter_goodbye
    - export
## Generated Story 8921121480760034253
* greet
    - utter_greet
* inform
    - utter_ask_location
* inform[location=London]
    - slot{"location": "London"}
    - action_weather
* goodbye
    - utter_goodbye
    - export
## Generated Story -5208991511085841103
```

Generated Story 3320800183399695936

- greet
  - utter_greet
- inform
  - utter_ask_location
- inform{"location": "italy"}
  - slot{"location": "italy"}
  - action_weather
  - slot{"location": "italy"}
- goodbye
  - utter_goodbye
  - export

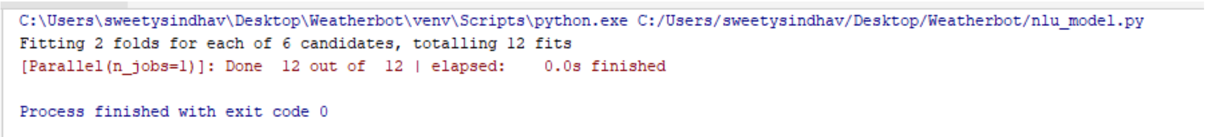
Generated Story -3351152636827275381

- greet
  - utter_greet

Fig: Generated Stories in a markdown file

7.3 Training the Model

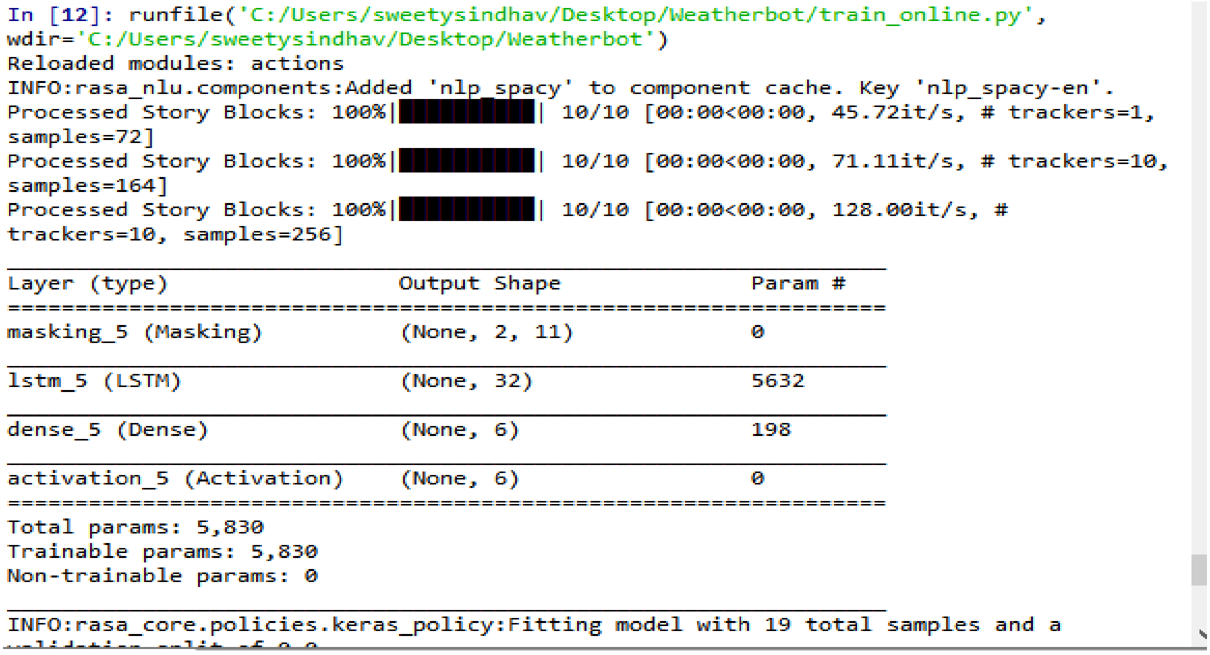
Intent Identification



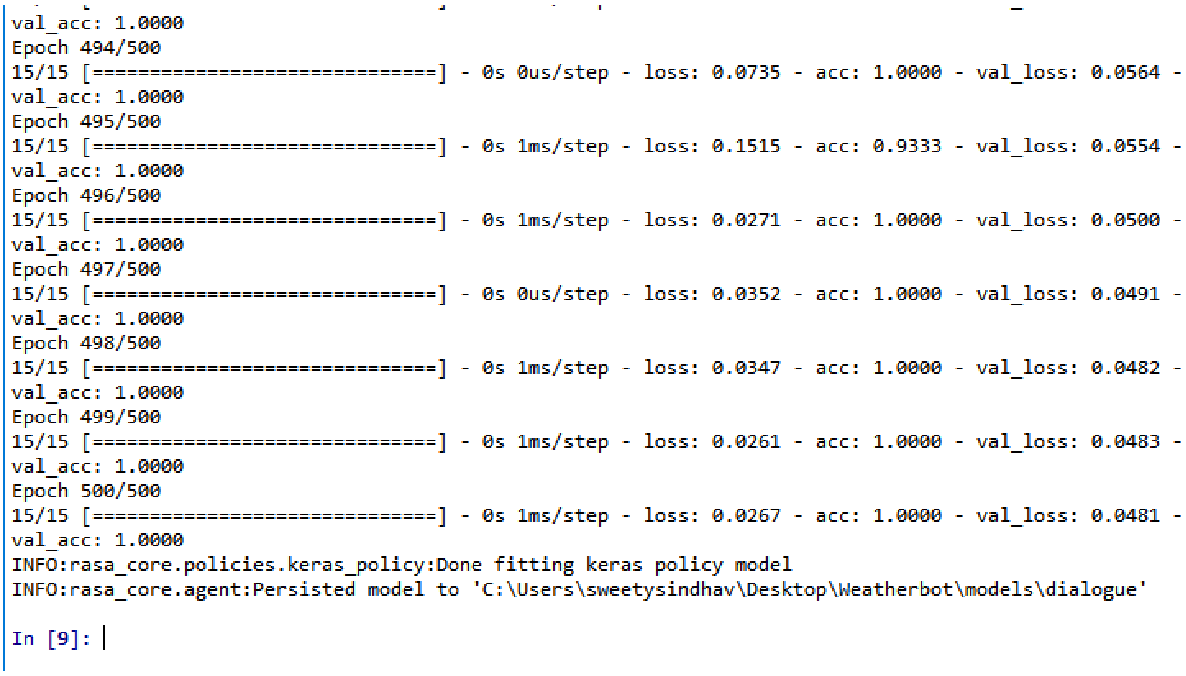
```
C:\Users\sweetysindhav\Desktop\Weatherbot\venv\Scripts\python.exe C:/Users/sweetysindhav/Desktop/Weatherbot/nlu_model.py
Fitting 2 folds for each of 6 candidades, totalling 12 fits
[Parallel(n_jobs=1)]: Done  12 out of  12 | elapsed:    0.0s finished

Process finished with exit code 0
```

Fig: Training dataset for Intent Identification

Entity Extraction and Slot filling



```
In [12]: runfile('C:/Users/sweetysindhav/Desktop/Weatherbot/train_online.py',
wdir='C:/Users/sweetysindhav/Desktop/Weatherbot')
Reloaded modules: actions
INFO:rasa_nlu.components:Added 'nlp_spacy' to component cache. Key 'nlp_spacy-en'.
Processed Story Blocks: 100%|██████████| 10/10 [00:00<00:00, 45.72it/s, # trackers=1,
samples=72]
Processed Story Blocks: 100%|██████████| 10/10 [00:00<00:00, 71.11it/s, # trackers=10,
samples=164]
Processed Story Blocks: 100%|██████████| 10/10 [00:00<00:00, 128.00it/s, #
trackers=10, samples=256]
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
masking_5 (Masking)          (None, 2, 11)             0
_________________________________________________________________
lstm_5 (LSTM)                (None, 32)                5632
_________________________________________________________________
dense_5 (Dense)              (None, 6)                 198
_________________________________________________________________
activation_5 (Activation)    (None, 6)                 0
=================================================================
Total params: 5,830
Trainable params: 5,830
Non-trainable params: 0
_________________________________________________________________
INFO:rasa_core.policies.keras_policy:Fitting model with 19 total samples and a
validation split of 0.0
```

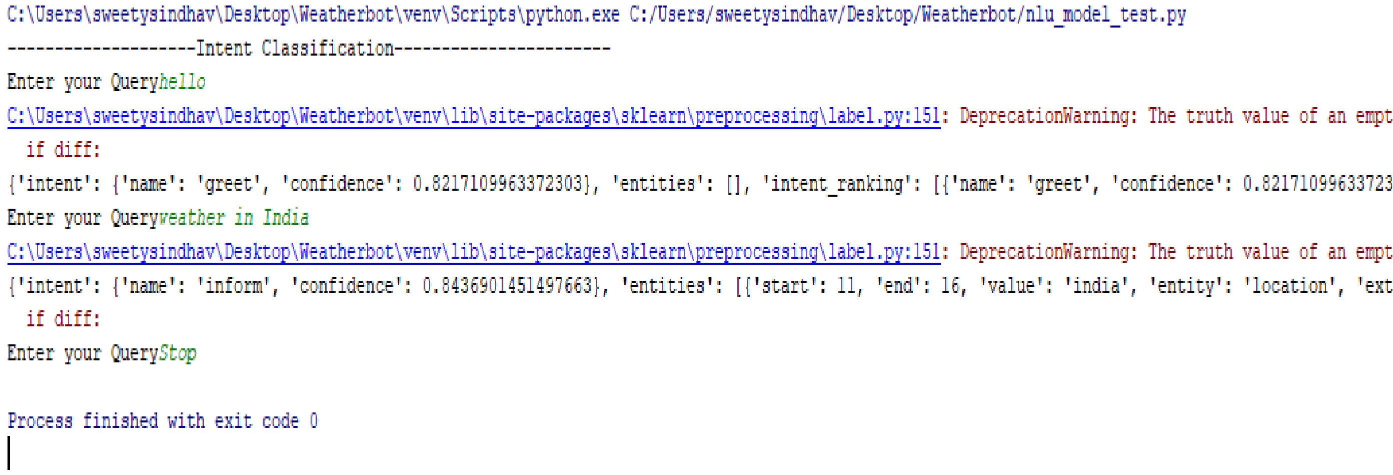


```
val_acc: 1.0000
Epoch 494/500
15/15 [==============================] - 0s 0us/step - loss: 0.0735 - acc: 1.0000 - val_loss: 0.0564 -
val_acc: 1.0000
Epoch 495/500
15/15 [==============================] - 0s 1ms/step - loss: 0.1515 - acc: 0.9333 - val_loss: 0.0554 -
val_acc: 1.0000
Epoch 496/500
15/15 [==============================] - 0s 1ms/step - loss: 0.0271 - acc: 1.0000 - val_loss: 0.0500 -
val_acc: 1.0000
Epoch 497/500
15/15 [==============================] - 0s 0us/step - loss: 0.0352 - acc: 1.0000 - val_loss: 0.0491 -
val_acc: 1.0000
Epoch 498/500
15/15 [==============================] - 0s 1ms/step - loss: 0.0347 - acc: 1.0000 - val_loss: 0.0482 -
val_acc: 1.0000
Epoch 499/500
15/15 [==============================] - 0s 1ms/step - loss: 0.0261 - acc: 1.0000 - val_loss: 0.0483 -
val_acc: 1.0000
Epoch 500/500
15/15 [==============================] - 0s 1ms/step - loss: 0.0267 - acc: 1.0000 - val_loss: 0.0481 -
val_acc: 1.0000
INFO:rasa_core.policies.keras_policy:Done fitting keras policy model
INFO:rasa_core.agent:Persisted model to 'C:\Users\sweetysindhav\Desktop\Weatherbot\models\dialogue'

In [9]: |
```

Fig : Training Model for slot filling and model persisted

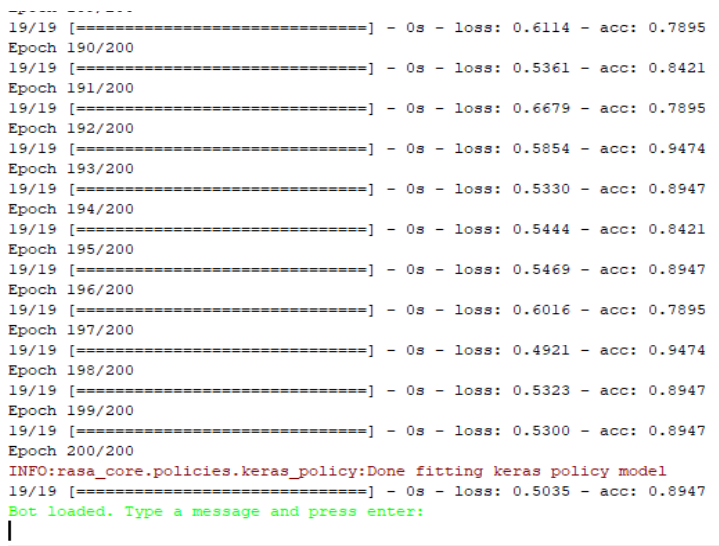
7.4 Testing the model

The model Classify the intent with its confidence score and also recongnize entity type and its value from the given text.

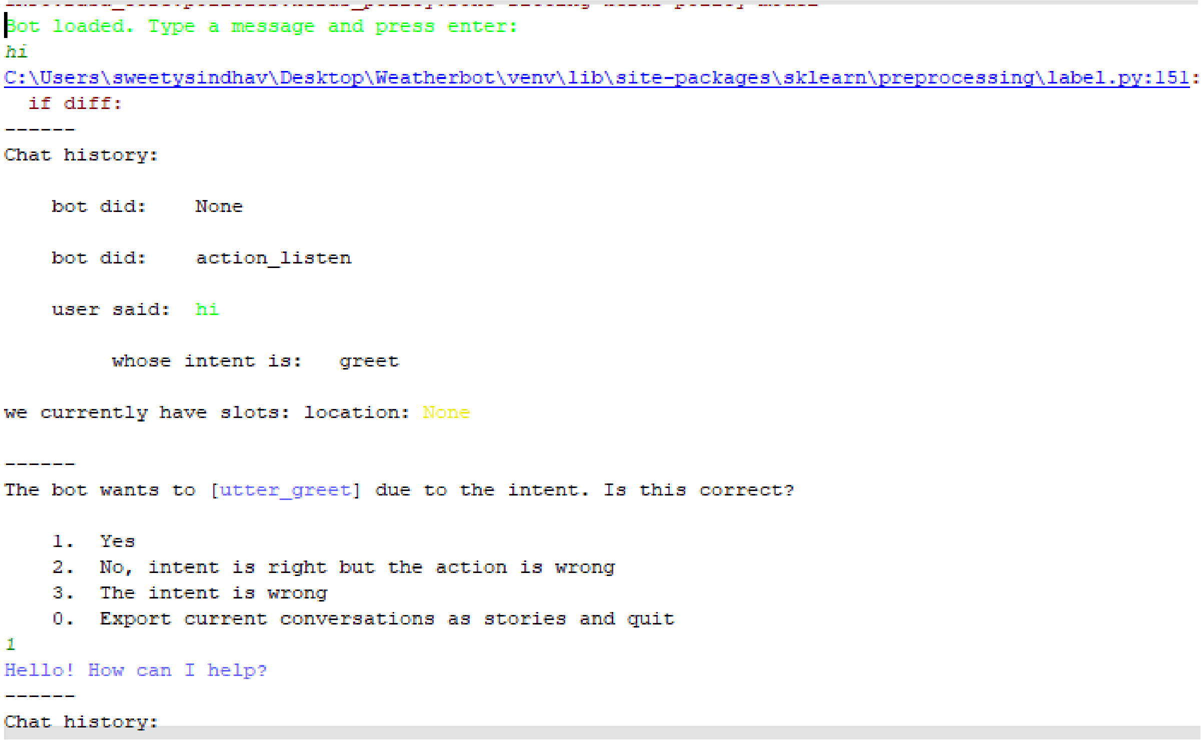


```
C:\Users\sweetysindhav\Desktop\Weatherbot\venv\Scripts\python.exe C:/Users/sweetysindhav/Desktop/Weatherbot/nlu_model_test.py
--------------------Intent Classification-----------------------
Enter your Queryhello
C:\Users\sweetysindhav\Desktop\Weatherbot\venv\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an empt
  if diff:
{'intent': {'name': 'greet', 'confidence': 0.8217109963372303}, 'entities': [], 'intent_ranking': [{'name': 'greet', 'confidence': 0.82171099633723
Enter your Queryweather in India
C:\Users\sweetysindhav\Desktop\Weatherbot\venv\lib\site-packages\sklearn\preprocessing\label.py:151: DeprecationWarning: The truth value of an empt
{'intent': {'name': 'inform', 'confidence': 0.8436901451497663}, 'entities': [{'start': 11, 'end': 16, 'value': 'india', 'entity': 'location', 'ext
  if diff:
Enter your QueryStop

Process finished with exit code 0
```

Fig: Intent Identification with Confidence Score and Entity Extraction



```
19/19 [==============================] - 0s - loss: 0.6114 - acc: 0.7895
Epoch 190/200
19/19 [==============================] - 0s - loss: 0.5361 - acc: 0.8421
Epoch 191/200
19/19 [==============================] - 0s - loss: 0.6679 - acc: 0.7895
Epoch 192/200
19/19 [==============================] - 0s - loss: 0.5854 - acc: 0.9474
Epoch 193/200
19/19 [==============================] - 0s - loss: 0.5330 - acc: 0.8947
Epoch 194/200
19/19 [==============================] - 0s - loss: 0.5444 - acc: 0.8421
Epoch 195/200
19/19 [==============================] - 0s - loss: 0.5469 - acc: 0.8947
Epoch 196/200
19/19 [==============================] - 0s - loss: 0.6016 - acc: 0.7895
Epoch 197/200
19/19 [==============================] - 0s - loss: 0.4921 - acc: 0.9474
Epoch 198/200
19/19 [==============================] - 0s - loss: 0.5323 - acc: 0.8947
Epoch 199/200
19/19 [==============================] - 0s - loss: 0.5300 - acc: 0.8947
Epoch 200/200
INFO:rasa_core.policies.keras_policy:Done fitting keras policy model
19/19 [==============================] - 0s - loss: 0.5035 - acc: 0.8947
Bot loaded. Type a message and press enter:
```



```
Bot loaded. Type a message and press enter:
hi
C:\Users\sweetysindhav\Desktop\Weatherbot\venv\lib\site-packages\sklearn\preprocessing\label.py:151:
  if diff:
------
Chat history:

    bot did:    None

    bot did:    action_listen

    user said:  hi

         whose intent is:   greet

we currently have slots: location: None

------
The bot wants to [utter_greet] due to the intent. Is this correct?

    1.  Yes
    2.  No, intent is right but the action is wrong
    3.  The intent is wrong
    0.  Export current conversations as stories and quit
1
Hello! How can I help?
------
Chat history:
```

8.Conclusion

From this study we can conclude the importance of Intent Identification,Entity Extraction and Slot filling using machine learning model. It also provides better results as compared to Rule based System by identifing correct intent, entity and slot. We aslo improve the model by adding the data on the go while user interacting so that the model can be improved by training. In future work, the chatbot can be trained on large amount of data and including more variety of intent and entiy. Also can improve by training the model on open domain.

9.References

1) Intent Detection and Slots Prompt in a Closed-Domain Chatbot: Amber Nigam, Prashik Sahare, Kushagra Pandya kydots.ai 2019 IEEE 13th International Conference on Semantic Computing (ICSC)

2) Feifei Dai,Chong Feng,Zhiqiang Wang: Intent Identification for knowledge based Question Answering

3) Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification in Chatbot: Muhammad Yusril Helmi Setyawan, Rolly Maulana Awangga, Safif Rafi Efendi

4) Observations of a New Chatbot Drawing Conclusions from Early Interactions with Users: Lisa N. Michaud Aspect Software Published by the IEEE Computer Society September/October 2018 1520-9202/18/$33.00 ©2018 IEEE

5) Bayan Abu Shawar and Eric Atwell, 2007 “Chatbots: Are they Really Useful?”

6) https://github.com/apixu/apixu-python.git

7) www.rasa.com