CHATBOT USING PYTHON

PHASE 2

2.1 Short Explanation :

Chatbots, also known as conversational agents, are designed with the help of AI (Artificial Intelligence) software.

They simulate aconversation (or a chat) with users in a natural language via messaging applications, websites, mobile apps, or phone.

A chatbot can be used anywhere a human is interacting with a computer system.

These are the areas where the fastest adoption is occurring:

1. Customer service

2. Sales and Marketing

Because it is economically available 24*7. The technology at the core of the rise of the chatbot is natural language processing (“NLP”).

2.2 Objective :

This system is developing chat bot based on android system so with the combination of Artificial Intelligent Knowledgeable database and virtual assistance. We can develop such chat bot which will make a conversion between human and machine and will satisfy the question raised by user.

2.3 Download and Installation :

pip3 install -U pip

- Virtual Environment Setup

Create and activate the virtual environment using the below commands.

python3 -m venv ./venv

- Activate the virtual environment:

.\venv\Scripts\activate

- Install Rasa open-source with the below command:
  pip install rasa
- make a directory and move to it

  mkdir test-chatbot && cd test-chatbot

Create the new project with rasa init command and start the conversation with the initial demo chatbot.

2.4 Steps :

Step 1: Clean the data. We removed non-English requests and requests with images. All the mentions were also removed in the training and testing data.

Step 2: Tokenize the data. We built a vocabulary of the most frequent 100K words in the conversations.

Step 3: Generate word-embedding features. We used the collected corpus to train word2vec models. Each word in the vocabulary was represented as a 640-dimension vector.

Step 4: Train LSTM networks. The input and output of LSTMs are vector representations of word sequences, with one word encoded or decoded at a time.

2.5 Testing and Findings :

In the beginning of our project, we wanted to test the previous version of our chatbot.

This was late in the fall and most of the first-year students were familiar with a lot of the answers our chatbot could provide.

We wanted to test this early version of the prototype to get input on what the chatbot could and could not answer in the future.

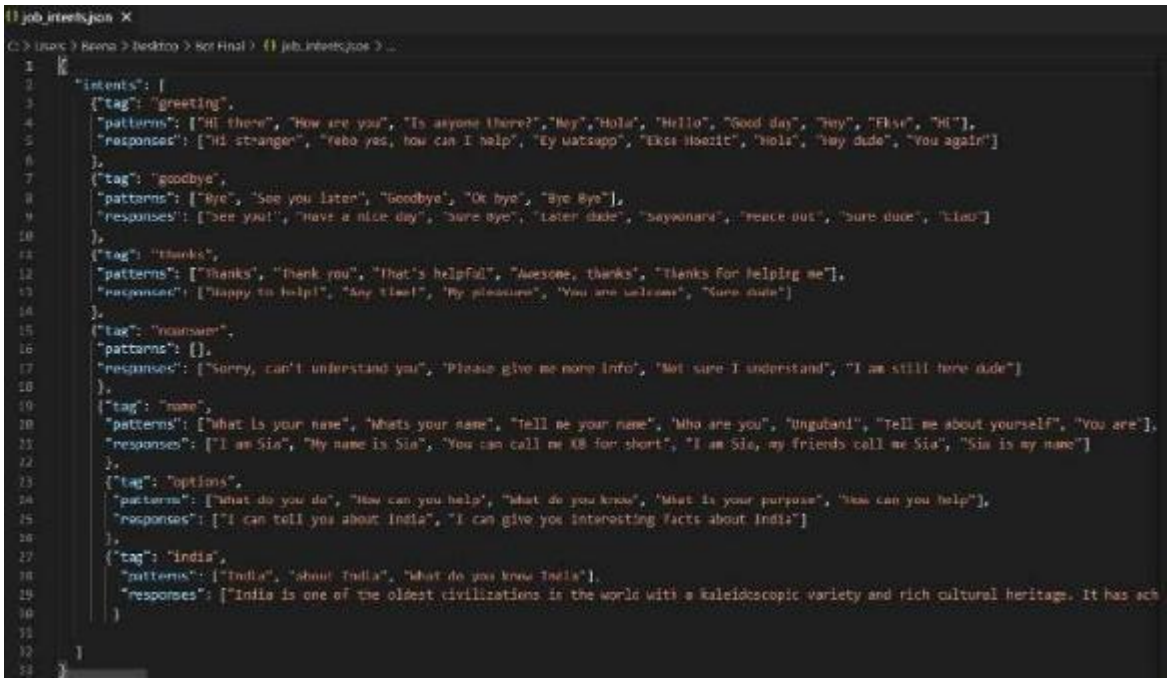
2.6 Performance Evaluation :

- Evaluate effectiveness.
- Measure ROI and costs.
- Gain customer satisfaction insights.
- Make data-driven decisions.

5. FUTURE SCOPE :

This project can be developed even more by adding multilanguages, speech recognition. We can add many more tags tothe data set, as the website gets developed. The chat history ofa particular user can be sent as a mail to him/her after theconversation is over. This can be done by authorizing the users and receiving their mail id's. This project is a smallinitiative to make the website user-friendly and easilyunderstandable by the user.

2.7 Data Set :



2.8 Implementation :

```
from __future__ import print_function
import numpy as np
import tensorflow as tf
 import argparse
 import os
 import pickle
import copy
 import sys
 import html  from utils
```

```
 import TextLoader from model

import Model  def main():

  assert sys.version_info >= (3, 3), \

   "Must be run in Python 3.3 or later. You are running
{}".format(sys.version)

  parser = argparse.ArgumentParser()

parser.add_argument('--save_dir', type=str, default='models/reddit',
help='model directory to store checkpointed models')
parser.add_argument('-n', type=int, default=500, help='number of
characters to sample')

 parser.add_argument('--prime', type=str, default=' ', help='prime text')
parser.add_argument('--beam_width', type=int, default=2,       help='Width
of the beam for beam search, default 2')

 parser.add_argument('--temperature', type=float, default=1.0,
help='sampling temperature'                  '(lower is more conservative,
default is 1.0, which is neutral)')

parser.add_argument('--topn', type=int, default=-1,                  help='at
each step, choose from only this many most likely characters;'
'set to <0 to disable top-n filtering.')

 parser.add_argument('--relevance', type=float, default=-1.,
help='amount of "relevance masking/MMI (disabled by default):"'
'higher is more pressure, 0.4 is probably as high as it can go without'
'noticeably degrading coherence;'                  'set to <0 to disable
relevance masking')raction skill.
```

Model.py

```
import tensorflow as tf from tensorflow.python.ops

import rnn_cell from tensorflow.python.ops

import nn_ops from tensorflow.python.ops

import variable_scope as vs from tensorflow.python.framework

import ops from tensorflow.contrib

import rnn  from tensorflow.python.util.nest
```

import flatten import numpy as np class PartitionedMultiRNNCell(rnn_cell.RNNCell): """RNN cell composed sequentially of multiple simple cells."""

2.9 Conclusion :

Chatbots are an extremely promising technology. They are bound to stay with us and will grow in popularity with the increasingadoption of messaging apps and newdigital communication channels. You should get

used to them as more and more companies are choosing chatbots for marketingpurposes and to automate customer service.