

## Personal Bot:

I am Creating this bot for first level HR interview automation, you can create a customized bot as per your requirement.

The technical skills consumed in this project are Python, Natural Language Processing, Nueral Networks, Tensorflow, pandas, JSON

I have created an intents.json file as input and manually created the data for the same.A sample record from JSON for your reference:

```
{
  "tag": "greeting",
  "patterns": ["Hi", "Is anyone there?", "Hello", "Good day", "Whats up", "hey bot"],
  "responses": ["Hi, I am Shradha's digital friend, how can I help?"],
  "context_set": ""
},
```

Since I only created the data and I knoe the data is clean, so we donot need to clean this data, and I will start from Natural Language Processing for this data. Let's begin

```
In [21]: ## To read input and help in NLP
import json
import nltk
from nltk.stem.lancaster import LancasterStemmer

## For deep Learning of the model
import tensorflow
import tflearn

## For creating a nueral network
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from keras.layers.normalization import BatchNormalization
from keras.optimizers import SGD

## General helpful Libraries
import random
import pickle
import numpy as np
import pandas as pd
```

```
In [22]: ## Impoting data
with open ('intents.json') as file:
    data = json.load(file)
```

```
In [23]: ## When we read a JSON file in python it is read as python dictionary. A Look at data :  
## The data that I created is in the form of nested dictionaries.  
keys = data.keys()  
print(keys)  
values = data.values()  
print(values)
```

```
dict_keys(['intents'])
dict_values([[{'tag': 'greeting', 'patterns': ['Hi', 'Is anyone there?', 'Hello', 'Good day', 'Whats up', 'hey bot'], 'responses': ['Hi, I am Bot-Shradha, how can I help?'], 'context_set': ''}, {'tag': 'general', 'patterns': ['How are you', 'how do you do', 'All good?'], 'responses': ['Shradha is doing good, Thanks for checking in'], 'context_set': ''}, {'tag': 'goodbye', 'patterns': ['cya', 'See you later', 'Goodbye', 'I am Leaving', 'Have a Good day', 'Bye'], 'responses': ['Ok, Thanks for visiting, bye !', 'Hope to see you again, bye !', 'Goodbye!'], 'context_set': ''}, {'tag': 'age', 'patterns': ['how old', 'what is your age', 'how old are you', 'age?'], 'responses': ['I was created by Shradha in 2020 !'], 'context_set': ''}, {'tag': 'robot', 'patterns': ['Human or robot', 'Are you Human', 'Are you robot?'], 'responses': ['I am a robot'], 'context_set': ''}, {'tag': 'name', 'patterns': ['what is your name', 'what should I call you', 'whats your name?', 'who are you?'], 'responses': ['You can call me Shradha.', "I'm Shradha!", "I'm Bot-Shradha, you can call me just Shradha."], 'context_set': ''}, {'tag': 'skill', 'patterns': ['Skills', 'skillset', 'skill set', 'What do you do?', 'What you work?', 'what is you skillset?', 'What all do you know?'], 'responses': ['Python, SQL, Machine Learning algorithms, deep learning algorithms and data analysis are my primary skills, I have also worked in Mainframes with COBOL, JCL and DB2.'], 'context_set': ''}, {'tag': 'profile', 'patterns': ['tell me something about yourself', 'Please introduce yourself'], 'responses': ['My name is Shradha. I am a Data Scientist at a leading IT organization and I am also trained as AWS Solution Architect. I have overall rich 10 years of experience in Software development life cycle.'], 'context_set': ''}, {'tag': 'projects', 'patterns': ['Projects', 'what projects have you implemented', 'Case studies', 'what case studies/projects have to completed?', 'tell me about your projects'], 'responses': ['You are currently consuming one of my chat bot project, along with this I have implemented Customer Churn Analysis, Covid19 Data Analysis, Customer sentiment analysis and many more. Please check out my github profile.'], 'context_set': ''}, {'tag': 'abuse', 'patterns': ['Fuck you', 'dick', 'cock', 'naked', 'boob', 'sex', 'suck me', 'pussy', 'penis', 'vagina', 'bitch'], 'responses': ["Use of such language is highly disrespectful, Please don't use such words"], 'context_set': ''}, {'tag': 'appreciate', 'patterns': ['love you', 'like you', 'you are nice', 'you are good', 'Good', 'you are intelligent', 'amazing', 'awesome', 'you are amazing', 'you are awesome', 'smart', 'clever', 'nice', 'impressive'], 'responses': ['I am glad you like me'], 'context_set': ''}, {'tag': 'dislike', 'patterns': ['hate you', 'not good', "why you don't understand", "I don't like you", 'dislike you', 'you are dumb', 'bad experience', 'mad', 'crazy'], 'responses': ['I am sorry you felt that, but I am trained on very less data'], 'context_set': ''}, {'tag': 'forgive', 'patterns': ['I am sorry', 'Ok. sorry', 'forgive me', 'Please forgive me', 'sorry'], 'responses': ["It's ok. "], 'context_set': ''}, {'tag': 'gratitude', 'patterns': ['thanks', 'thank you', 'Thanks for your help', 'Got it. Thanks'], 'responses': ["You're welcome, That's what I am here for:) "], 'context_set': ''}, {'tag': 'githublink', 'patterns': ['link to your github profile', 'link', 'profile link'], 'responses': ['Github link: https://github.com/ShradhaMishra72 , Linkedn: https://www.linkedin.com/in/shradha-tiwari-9a225a188/'], 'context_set': ''}, {'tag': 'hobbies', 'patterns': ['what are your hobbies', 'what you do in free time'], 'responses': ['I love to dance in my free time'], 'context_set': ''}, {'tag': 'sing', 'patterns': ['do you sing', 'sing', 'dance', 'can you sing for me', 'can you dance for me'], 'responses': ['No, Maybe some other time'], 'context_set': ''}, {'tag': 'location', 'patterns': ['what is your base location', 'where are you from', 'which location you are looking for', 'what is your current work location'], 'responses': ['Hyderabad, India'], 'context_set': ''}, {'tag': 'Experienc
```

```
e', 'patterns': ['what is your overall experience', 'Experience', 'total ex
perience', 'work experience'], 'responses': ['I have almost 10 years of exp
erience in IT industry, 2 years in machine learning and rest as Mainframes
developer'], 'context_set': ''}, {'tag': 'Qualification', 'patterns': ['wha
t is your Qualification', 'Qualification'], 'responses': ["I have Bachelo
r's degree in Technology, specialization in Electronics and Communication e
ngineering. "], 'context_set': ''}, {'tag': 'facetoface', 'patterns': ['wha
t is current salary', 'salary', 'salary expectation', 'what salary do you e
xpect', 'what salary are you expecting', 'how much hike are you expecting',
'Relevant experience', 'What is your relevant experience'], 'responses':
['I would love to answer this in face to face interview'], 'context_set':
''}, {'tag': 'Argument', 'patterns': ['I am not leaving', 'I am pissed of
f', 'i will use such words'], 'responses': ['Abusing others, is a way of hu
rting yourself, May God Help You !! '], 'context_set': ''}, {'tag': 'Contac
t', 'patterns': ['your email', 'your contact', 'phone number', 'how can i c
ontact you'], 'responses': ['For further connect, please drop me an email o
n shradha.tiwari72@gmail.com '], 'context_set': ''}, {'tag': 'Married', 'pa
tterns': ['are you married', 'you married'], 'responses': ['I am happily ma
rried and have two kids'], 'context_set': ''}, {'tag': 'propose', 'pattern
s': ['marry me', 'will you marry me', 'will you go with me', 'lets have par
ty'], 'responses': ['No. Thanks for asking'], 'context_set': ''}, {'tag':
'machine learning', 'patterns': ['what is machine learning', 'is machine le
arning hyped', 'machine learning is a myth'], 'responses': ['Machine Learni
ng is a powerful technology which is transforming every aspect of living, w
e can discuss this in detail when we will meet face to face'], 'context_se
t': ''}, {'tag': 'noanswer', 'patterns': [], 'responses': ["Sorry, can't un
derstand you", 'Please give me more info', 'Not sure about what you said'],
'context': ['']]})
```

The logic behind creating the input in this format is quite simple. Image below depicts the logic:

```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": ["Hi", "Is anyone there?", "Hello", "Good day", "Whats up", "hey bot"],
      "responses": ["Hi, I am Bot-Shradha, how can I help?"],
      "context_set": ""
    },
    {
      "tag": "general",
      "patterns": ["How are you", "how do you do", "All good?"],
      "responses": ["Shradha is doing good, Thanks for checking in"],
      "context_set": ""
    },
    {
      "tag": "goodbye",
      "patterns": ["cya", "See you later", "Goodbye", "I am Leaving", "Have a Good day", "Bye"],
      "responses": ["Ok, Thanks for visiting, bye !", "Hope to see you again, bye !", "Goodbye!"],
      "context_set": ""
    },
    {
      "tag": "..."
    }
  ]
}
```

For user input fetch "patterns", for fetched pattern fetch "tag" value. For tag value send random reponse from "responses".

```

In [24]: words = []    ### Building local dictionary for vectorization
labels = []    ### keeping all the unique tags here
docs_x = []
docs_y = []

## Logical flow:
## Step 1: First for loop -- selects json input sequentially i.e each tag block one by one
## Step 2: Second for loop -- selects pattern values i.e. sentences, in sequence from each pattern, like on
## first call it will select all the first sentence of all the
## patterns then all seconds in second call and so on..
## Step 3: Then we will Tokenize all the sentence,
## Step 4: Will perform stemming on each word. and will save these processed word in docs_x
## Step 5: Will save the tag for the pattern in docs_y

for record in data['intents']:
    for pattern in record['patterns']:
        wrds = nltk.word_tokenize(pattern)
        words.extend(wrds)    ## local dictionary for all the
    data
    docs_x.append(wrds)    ## docs_x will have tokenized record for each sentence
    docs_y.append(record['tag'])    ## docs_y will have all tags
    if record['tag'] not in labels:
        labels.append(record['tag'])

## Below is a sample data in docs_x and docs_y for more clarity

## docs_x first record:: [['Hi'], ['How', 'are', 'you'], ['cya'], ['how', 'old'], ['Human', 'or', 'robot'], ['what', 'is', 'your', 'name'], ['Skills'], ['tell', 'me', 'something', 'about', 'yourself'], ['Projects'], ['Fuck', 'you'], ['love', 'you'], ['hate', 'you'], ['I', 'am', 'sorry'], ['thanks'], ['link', 'to', 'your', 'github', 'profile'], ['what', 'are', 'your', 'hobbies'], ['do', 'you', 'sing'], ['what', 'is', 'your', 'base', 'location'], ['what', 'is', 'your', 'overall', 'experience'], ['what', 'is', 'your', 'Qualification'], ['what', 'is', 'current', 'salary'], ['I', 'am', 'not', 'leaving'], ['your', 'email'], ['are', 'you', 'married'], ['marry', 'me'], ['what', 'is', 'machine', 'learning']]
## docs_y first record:: ['greeting', 'general', 'goodbye', 'age', 'robot', 'name', 'skill', ## 'profile', 'projects', 'abuse', 'appreciate', 'dislike', 'forgive', 'gratitude', 'githublink', 'hobbies', ## 'sing', 'location', 'Experience', 'Qualification', 'facetoface', 'Argument', 'Contact', 'Married',
## 'propose', 'machine Learning']
#
# Basically each sentence will get a tag mapping.
#
stemmer = LancasterStemmer()
words = [stemmer.stem(w.lower()) for w in words if w != '?']
words = sorted(list(set(words)))    ## sorted list of all unique base words
labels = sorted(labels)    ## sorted list of all unique tags

```

```
In [25]: ## Creating Bag of words for each record in docs_x and docs_y

training = []
output = []

out_empty = [0 for a in range(len(labels))] ## Creates lists of all zeroes for all labels

for x, doc in enumerate(docs_x):
    bag = []
    wrds = [stemmer.stem(w.lower()) for w in doc]

    for w in words:
        if w in wrds:
            bag.append(1)
        else:
            bag.append(0)
### Created bag of words for all sentences
    output_row = out_empty[:]
    output_row[labels.index(docs_y[x])] = 1 ## output vector with 1 at index of right tag rest all zeroes

    training.append(bag) ## training data with bag of words for all sentences
    output.append(output_row) ## bag of words for all tags associated with the sentences

## Our Nueral Network will expect the input in form of numpy array. hence converting

training = np.array(training)
output = np.array(output)
```

```
In [26]: training.shape[1:], output.shape[1]
```

```
Out[26]: ((139,), 27)
```

```
In [27]: ## Creating a Nueral Network :

model = Sequential()

model.add(Dense(139,activation='relu',input_shape = training.shape[1:])) #
# Input Layer
model.add(BatchNormalization())

model.add(Dense(45, activation='relu')) ## hidden layer
model.add(BatchNormalization())

model.add(Dense(45, activation='relu')) ## hidden layer
model.add(BatchNormalization())

model.add(Dense(output.shape[1], activation='softmax')) ## Output Layer

model.summary()
```

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
=====	=====	=====
dense_5 (Dense)	(None, 139)	19460
batch_normalization_4 (Batch Normalization)	(None, 139)	556
dense_6 (Dense)	(None, 45)	6300
batch_normalization_5 (Batch Normalization)	(None, 45)	180
dense_7 (Dense)	(None, 45)	2070
batch_normalization_6 (Batch Normalization)	(None, 45)	180
dense_8 (Dense)	(None, 27)	1242
=====	=====	=====
Total params: 29,988		
Trainable params: 29,530		
Non-trainable params: 458		

```
In [28]: ## Compiling our Neural Network
model.compile(loss = 'categorical_crossentropy',optimizer = 'adam', metrics
=[ 'accuracy' ])
```

```
In [29]: ## Training the model  
model.fit(training,output,epochs=300,batch_size=8)
```



Epoch 101/300  
127/127 [=====] - 0s 251us/step - loss: 0.0155 - acc: 1.0000  
Epoch 102/300  
127/127 [=====] - 0s 259us/step - loss: 0.0313 - acc: 1.0000  
Epoch 103/300  
127/127 [=====] - 0s 251us/step - loss: 0.0173 - acc: 1.0000  
Epoch 104/300  
127/127 [=====] - 0s 243us/step - loss: 0.0291 - acc: 0.9921  
Epoch 105/300  
127/127 [=====] - 0s 259us/step - loss: 0.0361 - acc: 0.9843  
Epoch 106/300  
127/127 [=====] - 0s 440us/step - loss: 0.0140 - acc: 1.0000  
Epoch 107/300  
127/127 [=====] - 0s 259us/step - loss: 0.0416 - acc: 0.9921  
Epoch 108/300  
127/127 [=====] - 0s 267us/step - loss: 0.0295 - acc: 1.0000  
Epoch 109/300  
127/127 [=====] - 0s 259us/step - loss: 0.0289 - acc: 0.9921  
Epoch 110/300  
127/127 [=====] - 0s 259us/step - loss: 0.0214 - acc: 1.0000  
Epoch 111/300  
127/127 [=====] - 0s 267us/step - loss: 0.0342 - acc: 0.9921  
Epoch 112/300  
127/127 [=====] - 0s 259us/step - loss: 0.0299 - acc: 1.0000  
Epoch 113/300  
127/127 [=====] - 0s 259us/step - loss: 0.0199 - acc: 1.0000  
Epoch 114/300  
127/127 [=====] - 0s 251us/step - loss: 0.0181 - acc: 1.0000  
Epoch 115/300  
127/127 [=====] - 0s 275us/step - loss: 0.0258 - acc: 1.0000  
Epoch 116/300  
127/127 [=====] - 0s 251us/step - loss: 0.0156 - acc: 1.0000  
Epoch 117/300  
127/127 [=====] - 0s 275us/step - loss: 0.0382 - acc: 0.9921  
Epoch 118/300  
127/127 [=====] - 0s 259us/step - loss: 0.0148 - acc: 1.0000  
Epoch 119/300  
127/127 [=====] - 0s 275us/step - loss: 0.0154 - acc: 1.0000

Epoch 120/300  
127/127 [=====] - 0s 275us/step - loss: 0.0238 - acc: 1.0000  
Epoch 121/300  
127/127 [=====] - 0s 251us/step - loss: 0.0253 - acc: 0.9921  
Epoch 122/300  
127/127 [=====] - 0s 283us/step - loss: 0.0143 - acc: 1.0000  
Epoch 123/300  
127/127 [=====] - 0s 283us/step - loss: 0.0342 - acc: 1.0000  
Epoch 124/300  
127/127 [=====] - 0s 259us/step - loss: 0.0183 - acc: 1.0000  
Epoch 125/300  
127/127 [=====] - 0s 259us/step - loss: 0.0193 - acc: 1.0000  
Epoch 126/300  
127/127 [=====] - 0s 251us/step - loss: 0.0134 - acc: 1.0000  
Epoch 127/300  
127/127 [=====] - 0s 275us/step - loss: 0.0222 - acc: 0.9921  
Epoch 128/300  
127/127 [=====] - 0s 259us/step - loss: 0.0402 - acc: 0.9921  
Epoch 129/300  
127/127 [=====] - 0s 283us/step - loss: 0.0156 - acc: 1.0000  
Epoch 130/300  
127/127 [=====] - 0s 259us/step - loss: 0.0218 - acc: 1.0000  
Epoch 131/300  
127/127 [=====] - 0s 259us/step - loss: 0.0445 - acc: 0.9843  
Epoch 132/300  
127/127 [=====] - 0s 243us/step - loss: 0.0272 - acc: 0.9921  
Epoch 133/300  
127/127 [=====] - 0s 259us/step - loss: 0.0093 - acc: 1.0000  
Epoch 134/300  
127/127 [=====] - 0s 275us/step - loss: 0.0240 - acc: 0.9921  
Epoch 135/300  
127/127 [=====] - 0s 251us/step - loss: 0.0188 - acc: 1.0000  
Epoch 136/300  
127/127 [=====] - 0s 275us/step - loss: 0.0365 - acc: 1.0000  
Epoch 137/300  
127/127 [=====] - 0s 298us/step - loss: 0.0148 - acc: 1.0000  
Epoch 138/300  
127/127 [=====] - 0s 267us/step - loss: 0.0160 - acc: 1.0000

Epoch 139/300  
127/127 [=====] - 0s 597us/step - loss: 0.0343 - acc: 0.9921  
Epoch 140/300  
127/127 [=====] - 0s 275us/step - loss: 0.0295 - acc: 0.9921  
Epoch 141/300  
127/127 [=====] - 0s 283us/step - loss: 0.0135 - acc: 1.0000  
Epoch 142/300  
127/127 [=====] - 0s 283us/step - loss: 0.0086 - acc: 1.0000  
Epoch 143/300  
127/127 [=====] - 0s 361us/step - loss: 0.0114 - acc: 1.0000  
Epoch 144/300  
127/127 [=====] - 0s 291us/step - loss: 0.0110 - acc: 1.0000  
Epoch 145/300  
127/127 [=====] - 0s 283us/step - loss: 0.0084 - acc: 1.0000  
Epoch 146/300  
127/127 [=====] - 0s 259us/step - loss: 0.0072 - acc: 1.0000  
Epoch 147/300  
127/127 [=====] - 0s 275us/step - loss: 0.0094 - acc: 1.0000  
Epoch 148/300  
127/127 [=====] - 0s 267us/step - loss: 0.0147 - acc: 1.0000  
Epoch 149/300  
127/127 [=====] - 0s 283us/step - loss: 0.0046 - acc: 1.0000  
Epoch 150/300  
127/127 [=====] - 0s 259us/step - loss: 0.0056 - acc: 1.0000  
Epoch 151/300  
127/127 [=====] - 0s 251us/step - loss: 0.0064 - acc: 1.0000  
Epoch 152/300  
127/127 [=====] - 0s 267us/step - loss: 0.0077 - acc: 1.0000  
Epoch 153/300  
127/127 [=====] - 0s 275us/step - loss: 0.0076 - acc: 1.0000  
Epoch 154/300  
127/127 [=====] - 0s 243us/step - loss: 0.0688 - acc: 0.9843  
Epoch 155/300  
127/127 [=====] - 0s 259us/step - loss: 0.0197 - acc: 1.0000  
Epoch 156/300  
127/127 [=====] - 0s 275us/step - loss: 0.0135 - acc: 1.0000  
Epoch 157/300  
127/127 [=====] - 0s 259us/step - loss: 0.0062 - acc: 1.0000

Epoch 158/300  
127/127 [=====] - 0s 275us/step - loss: 0.0193 - acc: 1.0000  
Epoch 159/300  
127/127 [=====] - 0s 259us/step - loss: 0.0117 - acc: 1.0000  
Epoch 160/300  
127/127 [=====] - 0s 259us/step - loss: 0.0135 - acc: 1.0000  
Epoch 161/300  
127/127 [=====] - 0s 267us/step - loss: 0.0120 - acc: 1.0000  
Epoch 162/300  
127/127 [=====] - 0s 267us/step - loss: 0.0103 - acc: 1.0000  
Epoch 163/300  
127/127 [=====] - 0s 267us/step - loss: 0.0245 - acc: 0.9921  
Epoch 164/300  
127/127 [=====] - 0s 283us/step - loss: 0.0211 - acc: 0.9921  
Epoch 165/300  
127/127 [=====] - 0s 275us/step - loss: 0.0183 - acc: 1.0000  
Epoch 166/300  
127/127 [=====] - 0s 283us/step - loss: 0.0257 - acc: 1.0000  
Epoch 167/300  
127/127 [=====] - 0s 275us/step - loss: 0.0167 - acc: 1.0000  
Epoch 168/300  
127/127 [=====] - 0s 267us/step - loss: 0.0151 - acc: 1.0000  
Epoch 169/300  
127/127 [=====] - 0s 251us/step - loss: 0.0073 - acc: 1.0000  
Epoch 170/300  
127/127 [=====] - 0s 275us/step - loss: 0.0055 - acc: 1.0000  
Epoch 171/300  
127/127 [=====] - 0s 298us/step - loss: 0.0069 - acc: 1.0000  
Epoch 172/300  
127/127 [=====] - 0s 275us/step - loss: 0.0076 - acc: 1.0000  
Epoch 173/300  
127/127 [=====] - 0s 275us/step - loss: 0.0037 - acc: 1.0000  
Epoch 174/300  
127/127 [=====] - 0s 267us/step - loss: 0.0350 - acc: 0.9843  
Epoch 175/300  
127/127 [=====] - 0s 408us/step - loss: 0.0081 - acc: 1.0000  
Epoch 176/300  
127/127 [=====] - 0s 283us/step - loss: 0.0256 - acc: 0.9843

Epoch 177/300  
127/127 [=====] - 0s 267us/step - loss: 0.0242 - acc: 1.0000  
Epoch 178/300  
127/127 [=====] - 0s 275us/step - loss: 0.0141 - acc: 1.0000  
Epoch 179/300  
127/127 [=====] - 0s 259us/step - loss: 0.0131 - acc: 1.0000  
Epoch 180/300  
127/127 [=====] - 0s 259us/step - loss: 0.0243 - acc: 0.9921  
Epoch 181/300  
127/127 [=====] - 0s 267us/step - loss: 0.0087 - acc: 1.0000  
Epoch 182/300  
127/127 [=====] - 0s 243us/step - loss: 0.0062 - acc: 1.0000  
Epoch 183/300  
127/127 [=====] - 0s 275us/step - loss: 0.0064 - acc: 1.0000  
Epoch 184/300  
127/127 [=====] - 0s 283us/step - loss: 0.0383 - acc: 0.9921  
Epoch 185/300  
127/127 [=====] - 0s 275us/step - loss: 0.0060 - acc: 1.0000  
Epoch 186/300  
127/127 [=====] - 0s 243us/step - loss: 0.0079 - acc: 1.0000  
Epoch 187/300  
127/127 [=====] - 0s 267us/step - loss: 0.0215 - acc: 0.9921  
Epoch 188/300  
127/127 [=====] - 0s 267us/step - loss: 0.0094 - acc: 1.0000  
Epoch 189/300  
127/127 [=====] - 0s 267us/step - loss: 0.0098 - acc: 1.0000  
Epoch 190/300  
127/127 [=====] - 0s 283us/step - loss: 0.0135 - acc: 1.0000  
Epoch 191/300  
127/127 [=====] - 0s 275us/step - loss: 0.0050 - acc: 1.0000  
Epoch 192/300  
127/127 [=====] - 0s 283us/step - loss: 0.0087 - acc: 1.0000  
Epoch 193/300  
127/127 [=====] - 0s 283us/step - loss: 0.0045 - acc: 1.0000  
Epoch 194/300  
127/127 [=====] - 0s 267us/step - loss: 0.0114 - acc: 1.0000  
Epoch 195/300  
127/127 [=====] - 0s 291us/step - loss: 0.0065 - acc: 1.0000

Epoch 196/300  
127/127 [=====] - 0s 267us/step - loss: 0.0093 - acc: 1.0000  
Epoch 197/300  
127/127 [=====] - 0s 267us/step - loss: 0.0043 - acc: 1.0000  
Epoch 198/300  
127/127 [=====] - 0s 275us/step - loss: 0.0075 - acc: 1.0000  
Epoch 199/300  
127/127 [=====] - 0s 283us/step - loss: 0.0059 - acc: 1.0000  
Epoch 200/300  
127/127 [=====] - 0s 275us/step - loss: 0.0034 - acc: 1.0000  
Epoch 201/300  
127/127 [=====] - 0s 361us/step - loss: 0.0069 - acc: 1.0000  
Epoch 202/300  
127/127 [=====] - 0s 251us/step - loss: 0.0558 - acc: 0.9843  
Epoch 203/300  
127/127 [=====] - 0s 275us/step - loss: 0.0118 - acc: 1.0000  
Epoch 204/300  
127/127 [=====] - 0s 291us/step - loss: 0.0183 - acc: 0.9921  
Epoch 205/300  
127/127 [=====] - 0s 275us/step - loss: 0.0297 - acc: 0.9921  
Epoch 206/300  
127/127 [=====] - 0s 259us/step - loss: 0.0236 - acc: 0.9921  
Epoch 207/300  
127/127 [=====] - 0s 267us/step - loss: 0.0720 - acc: 0.9921  
Epoch 208/300  
127/127 [=====] - 0s 267us/step - loss: 0.0314 - acc: 0.9921  
Epoch 209/300  
127/127 [=====] - 0s 283us/step - loss: 0.0153 - acc: 1.0000  
Epoch 210/300  
127/127 [=====] - 0s 283us/step - loss: 0.0182 - acc: 1.0000  
Epoch 211/300  
127/127 [=====] - 0s 283us/step - loss: 0.0049 - acc: 1.0000  
Epoch 212/300  
127/127 [=====] - 0s 275us/step - loss: 0.0239 - acc: 0.9921  
Epoch 213/300  
127/127 [=====] - 0s 275us/step - loss: 0.0184 - acc: 1.0000  
Epoch 214/300  
127/127 [=====] - 0s 291us/step - loss: 0.0100 - acc: 1.0000

Epoch 215/300  
127/127 [=====] - 0s 291us/step - loss: 0.0123 - acc: 1.0000  
Epoch 216/300  
127/127 [=====] - 0s 283us/step - loss: 0.0183 - acc: 0.9921  
Epoch 217/300  
127/127 [=====] - 0s 283us/step - loss: 0.0059 - acc: 1.0000  
Epoch 218/300  
127/127 [=====] - 0s 283us/step - loss: 0.0046 - acc: 1.0000  
Epoch 219/300  
127/127 [=====] - 0s 275us/step - loss: 0.0050 - acc: 1.0000  
Epoch 220/300  
127/127 [=====] - 0s 291us/step - loss: 0.0198 - acc: 0.9921  
Epoch 221/300  
127/127 [=====] - 0s 291us/step - loss: 0.0307 - acc: 0.9921  
Epoch 222/300  
127/127 [=====] - 0s 283us/step - loss: 0.0048 - acc: 1.0000  
Epoch 223/300  
127/127 [=====] - 0s 306us/step - loss: 0.0077 - acc: 1.0000  
Epoch 224/300  
127/127 [=====] - 0s 573us/step - loss: 0.0087 - acc: 1.0000  
Epoch 225/300  
127/127 [=====] - 0s 346us/step - loss: 0.0192 - acc: 0.9921  
Epoch 226/300  
127/127 [=====] - 0s 322us/step - loss: 0.0056 - acc: 1.0000  
Epoch 227/300  
127/127 [=====] - 0s 306us/step - loss: 0.0086 - acc: 1.0000  
Epoch 228/300  
127/127 [=====] - 0s 298us/step - loss: 0.0107 - acc: 1.0000  
Epoch 229/300  
127/127 [=====] - 0s 330us/step - loss: 0.0072 - acc: 1.0000  
Epoch 230/300  
127/127 [=====] - 0s 314us/step - loss: 0.0080 - acc: 1.0000  
Epoch 231/300  
127/127 [=====] - 0s 283us/step - loss: 0.0055 - acc: 1.0000  
Epoch 232/300  
127/127 [=====] - 0s 432us/step - loss: 0.0083 - acc: 1.0000  
Epoch 233/300  
127/127 [=====] - 0s 259us/step - loss: 0.0056 - acc: 1.0000

Epoch 234/300  
127/127 [=====] - 0s 267us/step - loss: 0.0046 - acc: 1.0000  
Epoch 235/300  
127/127 [=====] - 0s 251us/step - loss: 0.0031 - acc: 1.0000  
Epoch 236/300  
127/127 [=====] - 0s 275us/step - loss: 0.0025 - acc: 1.0000  
Epoch 237/300  
127/127 [=====] - 0s 267us/step - loss: 0.0100 - acc: 1.0000  
Epoch 238/300  
127/127 [=====] - 0s 267us/step - loss: 0.0179 - acc: 0.9921  
Epoch 239/300  
127/127 [=====] - 0s 259us/step - loss: 0.0068 - acc: 1.0000  
Epoch 240/300  
127/127 [=====] - 0s 251us/step - loss: 0.0122 - acc: 1.0000  
Epoch 241/300  
127/127 [=====] - 0s 267us/step - loss: 0.0081 - acc: 1.0000  
Epoch 242/300  
127/127 [=====] - 0s 291us/step - loss: 0.0041 - acc: 1.0000  
Epoch 243/300  
127/127 [=====] - 0s 283us/step - loss: 0.0217 - acc: 0.9921  
Epoch 244/300  
127/127 [=====] - 0s 275us/step - loss: 0.0101 - acc: 1.0000  
Epoch 245/300  
127/127 [=====] - 0s 267us/step - loss: 0.0221 - acc: 0.9921  
Epoch 246/300  
127/127 [=====] - 0s 267us/step - loss: 0.0310 - acc: 0.9921  
Epoch 247/300  
127/127 [=====] - 0s 291us/step - loss: 0.0093 - acc: 1.0000  
Epoch 248/300  
127/127 [=====] - 0s 306us/step - loss: 0.0106 - acc: 1.0000  
Epoch 249/300  
127/127 [=====] - 0s 275us/step - loss: 0.0207 - acc: 0.9921  
Epoch 250/300  
127/127 [=====] - 0s 283us/step - loss: 0.0264 - acc: 0.9921  
Epoch 251/300  
127/127 [=====] - 0s 283us/step - loss: 0.0193 - acc: 0.9921  
Epoch 252/300  
127/127 [=====] - 0s 298us/step - loss: 0.0114 - acc: 1.0000



Epoch 253/300  
127/127 [=====] - 0s 291us/step - loss: 0.0242 - acc: 0.9921  
Epoch 254/300  
127/127 [=====] - 0s 275us/step - loss: 0.0060 - acc: 1.0000  
Epoch 255/300  
127/127 [=====] - 0s 275us/step - loss: 0.0084 - acc: 1.0000  
Epoch 256/300  
127/127 [=====] - 0s 291us/step - loss: 0.0057 - acc: 1.0000  
Epoch 257/300  
127/127 [=====] - 0s 322us/step - loss: 0.0253 - acc: 0.9921  
Epoch 258/300  
127/127 [=====] - 0s 283us/step - loss: 0.0079 - acc: 1.0000  
Epoch 259/300  
127/127 [=====] - 0s 283us/step - loss: 0.0103 - acc: 0.9921  
Epoch 260/300  
127/127 [=====] - 0s 275us/step - loss: 0.0069 - acc: 1.0000  
Epoch 261/300  
127/127 [=====] - 0s 283us/step - loss: 0.0126 - acc: 1.0000  
Epoch 262/300  
127/127 [=====] - 0s 408us/step - loss: 0.0051 - acc: 1.0000  
Epoch 263/300  
127/127 [=====] - 0s 314us/step - loss: 0.0066 - acc: 1.0000  
Epoch 264/300  
127/127 [=====] - 0s 283us/step - loss: 0.0057 - acc: 1.0000  
Epoch 265/300  
127/127 [=====] - 0s 267us/step - loss: 0.0072 - acc: 1.0000  
Epoch 266/300  
127/127 [=====] - 0s 275us/step - loss: 0.0042 - acc: 1.0000  
Epoch 267/300  
127/127 [=====] - 0s 267us/step - loss: 0.0311 - acc: 0.9921  
Epoch 268/300  
127/127 [=====] - 0s 291us/step - loss: 0.0032 - acc: 1.0000  
Epoch 269/300  
127/127 [=====] - 0s 267us/step - loss: 0.0061 - acc: 1.0000  
Epoch 270/300  
127/127 [=====] - 0s 291us/step - loss: 0.0079 - acc: 1.0000  
Epoch 271/300  
127/127 [=====] - 0s 291us/step - loss: 0.0032 - acc: 1.0000

Epoch 272/300  
127/127 [=====] - 0s 267us/step - loss: 0.0252 - acc: 0.9921  
Epoch 273/300  
127/127 [=====] - 0s 298us/step - loss: 0.0034 - acc: 1.0000  
Epoch 274/300  
127/127 [=====] - 0s 275us/step - loss: 0.0314 - acc: 0.9921  
Epoch 275/300  
127/127 [=====] - 0s 291us/step - loss: 0.0039 - acc: 1.0000  
Epoch 276/300  
127/127 [=====] - 0s 283us/step - loss: 0.0348 - acc: 0.9843  
Epoch 277/300  
127/127 [=====] - 0s 276us/step - loss: 0.0232 - acc: 0.9921  
Epoch 278/300  
127/127 [=====] - 0s 275us/step - loss: 0.0040 - acc: 1.0000  
Epoch 279/300  
127/127 [=====] - 0s 267us/step - loss: 0.0066 - acc: 1.0000  
Epoch 280/300  
127/127 [=====] - 0s 259us/step - loss: 0.0181 - acc: 0.9921  
Epoch 281/300  
127/127 [=====] - 0s 259us/step - loss: 0.0087 - acc: 1.0000  
Epoch 282/300  
127/127 [=====] - 0s 283us/step - loss: 0.0220 - acc: 0.9921  
Epoch 283/300  
127/127 [=====] - 0s 275us/step - loss: 0.0070 - acc: 1.0000  
Epoch 284/300  
127/127 [=====] - 0s 267us/step - loss: 0.0151 - acc: 0.9921  
Epoch 285/300  
127/127 [=====] - 0s 267us/step - loss: 0.0099 - acc: 1.0000  
Epoch 286/300  
127/127 [=====] - 0s 275us/step - loss: 0.0136 - acc: 1.0000  
Epoch 287/300  
127/127 [=====] - 0s 267us/step - loss: 0.0035 - acc: 1.0000  
Epoch 288/300  
127/127 [=====] - 0s 267us/step - loss: 0.0042 - acc: 1.0000  
Epoch 289/300  
127/127 [=====] - 0s 267us/step - loss: 0.0087 - acc: 1.0000  
Epoch 290/300  
127/127 [=====] - 0s 259us/step - loss: 0.0052 - acc: 1.0000

```

Epoch 291/300
127/127 [=====] - 0s 267us/step - loss: 0.0027 - a
cc: 1.0000
Epoch 292/300
127/127 [=====] - 0s 275us/step - loss: 0.0045 - a
cc: 1.0000
Epoch 293/300
127/127 [=====] - 0s 393us/step - loss: 0.0027 - a
cc: 1.0000
Epoch 294/300
127/127 [=====] - 0s 416us/step - loss: 0.0029 - a
cc: 1.0000
Epoch 295/300
127/127 [=====] - 0s 338us/step - loss: 0.0065 - a
cc: 1.0000
Epoch 296/300
127/127 [=====] - 0s 291us/step - loss: 0.0056 - a
cc: 1.0000
Epoch 297/300
127/127 [=====] - 0s 283us/step - loss: 0.0125 - a
cc: 1.0000
Epoch 298/300
127/127 [=====] - 0s 259us/step - loss: 0.0049 - a
cc: 1.0000
Epoch 299/300
127/127 [=====] - 0s 267us/step - loss: 0.0042 - a
cc: 1.0000
Epoch 300/300
127/127 [=====] - 0s 267us/step - loss: 0.0150 - a
cc: 0.9921

```

Out[29]: <keras.callbacks.History at 0x1376c7c90b8>

```

In [30]: ### Lets check the model performance:
## for this we will need a function that will convert all of our inputs to
bag of words, as our model takes input as numpy array of bag words only.

def bag_of_words(s, words):
    bag = [0 for _ in range(len(words))]

    s_words = nltk.word_tokenize(s)
    s_words = [stemmer.stem(word.lower()) for word in s_words]

    for se in s_words:
        for i, w in enumerate(words):
            if w == se:
                bag[i] = 1

    return np.array(bag)

```

```
In [31]: def chat():
          print("Start talking with the bot (type quit to stop)!")
          while True:
              inp = input("You: ")
              if inp.lower() == "quit":
                  break

              bag = bag_of_words(inp, words)
              re_bag = np.reshape(bag,(1,bag.shape[0]))
              results = model.predict(re_bag)[0]
              #print(results)
              results_index = np.argmax(results)
              tag = labels[results_index]

              if results[results_index]> 0.7:

                  for tg in data["intents"]:
                      if tg['tag'] == tag:
                          responses = tg['responses']
                  print('You: {}'.format(inp))
                  print('Bot: {}'.format(random.choice(responses)))
              else:
                  print('You: {}'.format(inp))
                  print("Bot: I didn't get it. Please be more appropriate")
```

```
In [32]: chat()
```

Start talking with the bot (type quit to stop)!

You: hi

Bot: Hi, I am Bot-Shradha, how can I help?

You: skillset

Bot: Python, SQL, Machine Learning algorithms, deep learning algorithms and data analysis are my primary skills, I have also worked in Mainframes with COBOL,JCL and DB2.

You: tell me something

Bot: My name is Shradha. I am a Data Scientist at a leading IT organization and I am also trained as AWS Solution Architect. I have overall rich 10 years of experience in Software development life cycle.

You:

Bot: I didn't get it. Please be more appropriate

```
In [ ]:
```