

# Almost Human

Member's Name : Humayan Ahmmed

Md. Faisal Ahmed

Member's Email : [humayan.ahmmed@gmail.com](mailto:humayan.ahmmed@gmail.com)  
[faisalrtlsnk@gmail.com](mailto:faisalrtlsnk@gmail.com)

## Introduction:

The aim of this project is to build an intelligent chatbot for an online based coffee shop. It mimics an actual person and interacts with users using text autonomously. We made it based on English language and it can provide decent reply to the users.

## Motivation

The purpose of this project is to showcase the power of chatbots and how they can be an alternative to using an application or even a website. The chatbots should be easy to use, respond in a timely fashion and be all round user friendly. The bots should make the users interaction as easy and fast as possible to ensure that the users time is not wasted and that they get what they want without any difficulty or misunderstanding from the bot. The conversation should flow and always keep the user in control of the conversation. Users should come away from their experience with the chatbot and think that it was a fun, easy to use and straightforward interaction that would encourage them to come back without any hesitation. With messaging platforms being the most used type of application in the world, businesses will be looking to take advantage of this and start to develop their own chatbots to work along with their social media pages. For example, a person calling a restaurant to see what time they open at or what is the special today, the customer can simply message the page on Facebook and the bot will respond accordingly. This frees up time for real

employees to do other work and allows the chatbot to handle the simple tasks.

## **Challenges**

The most challenging part was to choose the proper hyperparameters (batch size, number of units in hidden layers, activation functions, learning rate etc) for our model. There are various ways to implement a chatbot, among them select a particular way was also difficult for us. Although, some part of the dataset was taken from github, we created most of the parts of our dataset.

## **Related Works:**

Recently, chatbots received an increased attention from industry and diverse research communities as a dialogue-based interface providing advanced human-computer interactions. On the other hand, Open Data continues to be an important trend and a potential enabler for government transparency and citizen participation. [1] This paper shows how these two paradigms can be combined to help non-expert users find and discover open government datasets through dialogue.

[2] In this paper, a group of Amazon researchers propose a system that leverages user-system interaction feedback signals to automate learning without any manual annotation. Users here tend to modify a previous query in hopes of fixing an error in the previous turn to get the right results. In some cases, users may not properly formulate their requests (e.g. providing partial title of a song), but gleaning across a wider pool of users and sessions reveals the underlying recurrent patterns. The proposed self-learning system automatically detects the errors, generate reformulations and deploys fixes to the runtime system to correct different types of errors occurring in different components of the system.

Neural dialogue models have been widely adopted in various chatbot applications because of their good performance in simulating and generalizing human conversations. However, there exists a dark side of these models — due to the vulnerability of neural networks, a neural dialogue model can be manipulated by users to say what they want, which brings in concerns about the security of practical chatbot services.[3] This paper investigates whether we can craft inputs that lead a well-trained black-box neural dialogue model to generate targeted outputs. Experiments conducted on a representative neural dialogue model show that the proposed model is able to discover such desired inputs in a considerable portion of cases.

We think that improving the dataset can improve the performance of the chatbot. So, the next challenge can be to improve the dataset. Another challenge is to improve the hyperparameters.

## **Project Objectives:**

In this section, we will discuss the objectives of the project. We have implemented a chatbot for a coffee shop where customers can have conversation with the bot. A customer can talk on various things like availability of the products, payment policy, delivery etc. This chatbot can answer these questions decently.

### **Task of our system**

1. NLP concepts which includes stemming, tokenization, bag of words.

# NPL processing pipeline

"Is anyone there?"

↓ **tokenize**

["Is", "anyone", "there", "?"]

↓ **lower+stem**

["is", "anyone", "there", "?"]

↓ **exclude punctuation characters**

["is", "anyone", "there"]

↓ **bag of words**

**X** [0,0,0,1,0,1,0,1]

2. Create training data.
3. PyTorch modeling and training.
4. Save model and implementation of the chat.

## Some dummy inputs and outputs of chatbot:

```
Let's chat! (type 'quit' to exit)
You: Hi
Almost Human: Hi there, how can I help?
You: Hi
Almost Human: Hi there, what can I do for you?
You: What do you sell ?
Almost Human: We have coffee and tea
You: How long takes shipping ?
Almost Human: Delivery takes 2-4 days
You: Can you help me ?
Almost Human: Sorry, I don't understand....
You: Crack a joke
Almost Human: What did the buffalo say when his son left for college? Bison.
You: Bye
Almost Human: Have a nice day
You: quit
```

## Methodologies:

In our dataset, each conversational intent contains:

1. a tag (a unique name)
2. patterns (sentence patterns for our neural network text classifier)
3. responses (one will be used as a response)

At first, we have performed NLP tokenization. All the sentences in pattern section under each tag are tokenized. All the tags are also stored.

Secondly, we have excluded all the punctuation letters(' ', '!', '!', '?') from the tokenized words. Then, we have converted all the tokenized words in lowercase letters.

After that, we have applied NLP stemming on these tokenized words and sorted them to get unique words.

Next, we have created the bag of words. By using unique tokenized words as well as pattern sentences of each tag we have created our bag of words. It is our training data(X\_train). We have given all the tags a label like "greetings" tag gets label 0. It is our Y\_train.

Next, we have implemented our model, it is a 3 layer feed forward neural network. Input layer gets the bag of words as input.

First two layers use relu activation function and output layer uses softmax activation function. Using X\_train and Y\_train we have trained our model. For loss function we have selected cross-entropy loss function.

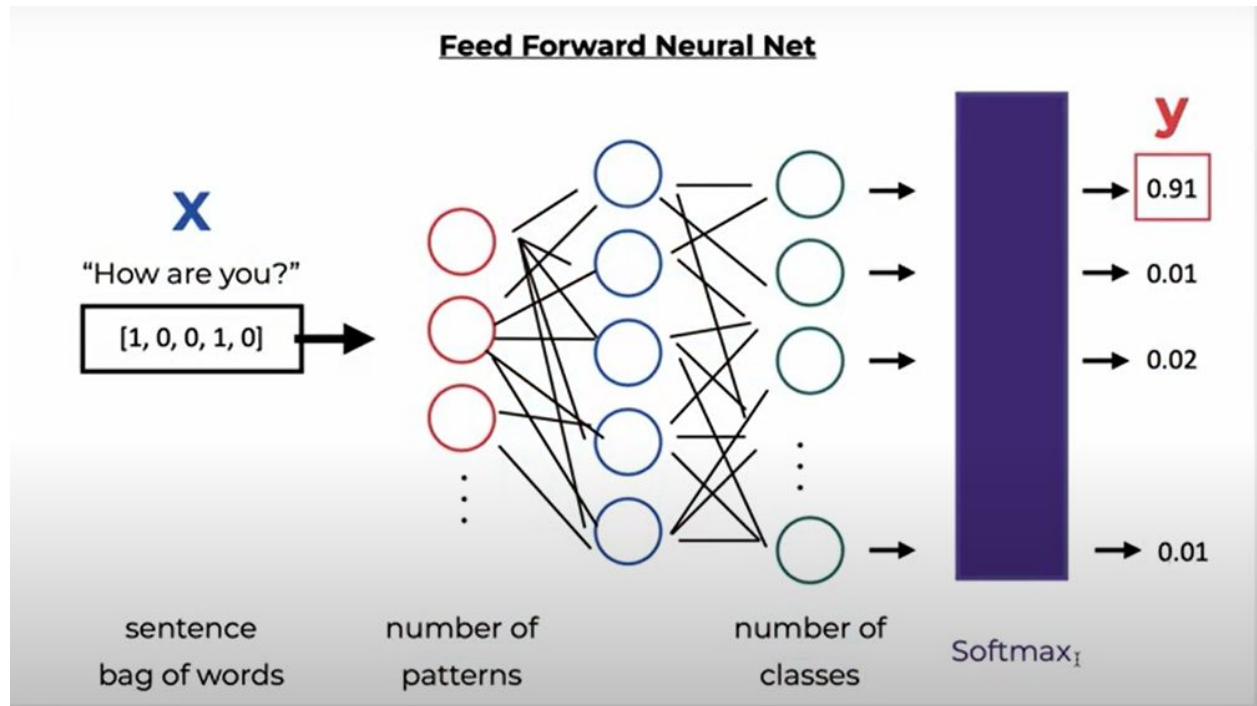


Fig. Feed forward Neural Net of our project.

## Experiment:

### Dataset

The dataset is a json file which have total 13 tags/classes. Each tag has some patterns and corresponding responses.

### A portion of our dataset:

```
"tag": "greeting",
  "patterns": [
    "Hi",
    "Hey",
    "How are you",
    "Is anyone there?",
    "Hello",
    "Good day"
  ],
```

```
"responses": [  
    "Hey :-)",  
    "Hello, thanks for visiting",  
    "Hi there, what can I do for you?",  
    "Hi there, how can I help?"  
]
```

## Evaluation Matrix

Difference in performance due to change in learning rate

Learning rate	Loss
0.001	0.0001
0.0001	1.0431
0.00001	2.5046

Difference in performance due to change in number of Epochs

Number of Epochs	Loss
800	0.0001
700	0.0004
600	0.0003

## Result

We have trained our model considering various scenarios that can occurs between a customer and the chatbot. The output has been satisfactory in most cases. It can be improved with much better dataset.

**Conclusions:** Arter evaluation we can see that the chatbot response pretty much well. Although there are many spaces for improvements. With proper dataset and improved hyper parameters we believe that the

performance of the chatbot can be improved. In future we will try to focus to improve this cases.

## **References:**

1. <https://arxiv.org/pdf/1909.03653.pdf>
2. <https://arxiv.org/pdf/1911.02557.pdf>
3. <https://arxiv.org/pdf/1909.06044.pdf>