# CHATBOT USING PYTORCH

Deployment: (ml-chatbot.vercel.app)

Rohit Kumar Sah
Department of Computer Science
Amrita School of Engineering
Chennai, Tamilnadu

G. Amrutha
Department of Computer Science
Amrita School of Engineering
Chennai, Tamilnadu

Birat Chapagain
Department of Computer Science
Amrita School of Engineering
Chennai, Tamilnadu

A.S.L. Manasa
Department of Computer Science
Amrita School of Engineering
Chennai, Tamilnadu

Inti Vamsi Priyanka
Department of Computer Science
Amrita School of Engineering
Chennai Tamilnadu

*Abstract*— **Chatbots, or conversational interfaces as they are also known, present a new way for individuals to interact with computer systems. Traditionally, to get a question answered by a software program involved using a search engine, or filling out a form. A chatbot allows a user to simply ask questions in the same manner that they would address a human. The most well-known chatbots currently are voice chatbots: Alexa and Siri. However, chatbots are currently being adopted at a high rate on computer chat platforms.**

**The technology at the core of the rise of the chatbot is natural language processing ("NLP"). Recent advances in machine learning have greatly improved the accuracy and effectiveness of natural language processing, making chatbots a viable option for many organizations. This improvement in NLP is firing a great deal of additional research which should lead to continued improvement in the effectiveness of chatbots in the years to come.**

***Keywords—NLP, feed-forward, chatbot, RNN etc.***

## I. Introduction

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person. You've probably interacted with chatbot whether you know it or not. For example, you're at your computer researching a product, and a window pops up on your screen asking if you need help. Or perhaps you're on your way to a concert and you use your smartphone to request a ride via chat. The working of chatbots is driven by AI, automated rules, natural-language processing (NLP), and machine learning (ML), chatbots process data to deliver responses to requests of all kinds.

There are two main types of chatbots are Task-oriented (declarative) chatbots and Data-driven and predictive (conversational) chatbots. The former type of chatbots includes single-purpose programs that focus on performing one function. Using rules, NLP, and very little ML, they generate automated but conversational responses to user inquiries. Interactions with these chatbots are highly specific and structured and are most applicable to support and service functions—think robust interactive FAQs. The latter type of chatbots are referred to as virtual assistants or digital assistants, and they are much more sophisticated, interactive, and personalized than task-oriented chatbots. These chatbots are contextually aware and leverage natural-language understanding (NLU), NLP, and ML to learn as they go. They apply predictive intelligence and analytics to enable personalization based on user profiles and past user behavior.

Apple's Siri and Amazon's Alexa are examples of consumer-oriented, data-driven, predictive chatbots. Chatbots boost operational efficiency and bring cost savings to businesses while offering convenience and added services to internal employees and external customers. They allow companies to easily resolve many types of customer queries and issues while reducing the need for human interaction. Soon, when AI is combined with the development of 5G technology, businesses, employees, and consumers are likely to enjoy enhanced chatbot features such as faster recommendations and predictions, and easy access to high-definition video conferencing from within a conversation. Our chatbot is designed in a way such that we have defined certain frequently asked questions(FAQs) and trained our chatbot with this data using neural networks, NLP and PyTorch. Proceeding further in the paper we will understand how the chatbot is designed and executed.

## II. Related Work

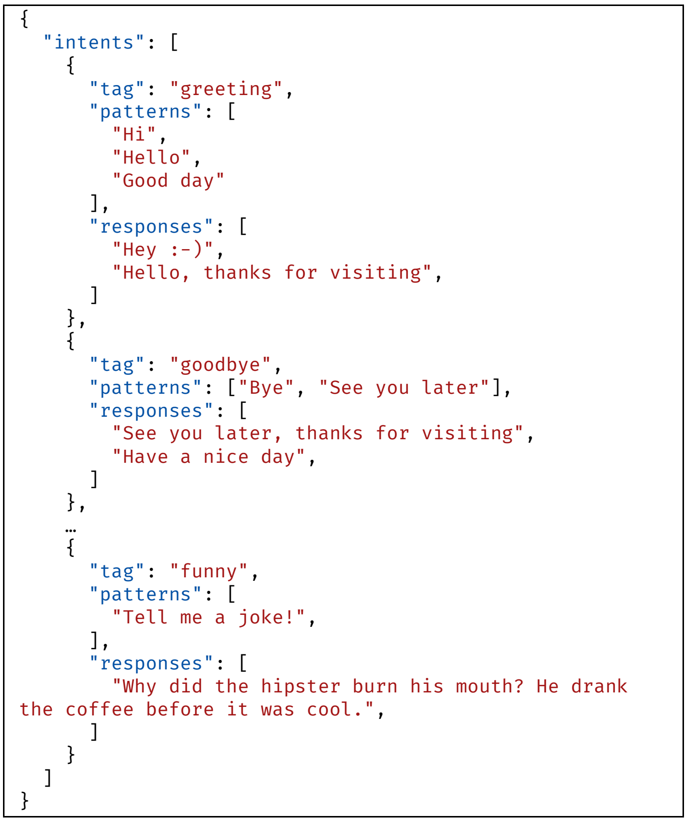
A chatbot is a computer program that simulates human conversation through voice commands or text chats or both. Chatbot, short for chatterbot, is an artificial intelligence (AI) feature that can be embedded and used through any major messaging application. There are many synonyms for chatbot, including "talkbot," "bot," "IM bot," "interactive agent" or "artificial conversation entity."

Today, most large-scale conversational AI agents (e.g. Alexa, Siri, or Google Assistant) are built using manually annotated data to train the different components of the system. Typically, the accuracy of the ML models in these components is improved by manually transcribing and annotating data. As the scope of these systems increases to cover more scenarios and domains, manual annotation to improve the accuracy of these components becomes prohibitively costly and time-consuming.

Work on retrieval-based chatbots, like most sequence pair matching tasks, can be divided into Cross-encoders that perform word matching over the pair, and Bi-encoders that encode the pair separately. The latter has better performance, however since candidate responses cannot be encoded offline, it is also much slower. Lately, multi-layer transformer architectures pre-trained as language models have been used to great effect on a variety of natural language processing and information retrieval tasks. Some of the innovative chatbots are Endurance, Casper, Disney, etc.

## III. Dataset

The dataset is made up of WikiQA and smaller query response datasets. It has around 2800 intents, each with its own unique tag and a set of patterns (queries) and replies (answers). The dataset's tags enable the model to recognize trends and responses. The pattern specifies the sort of question that can be asked in response to the same query. Because many questions might have the same meaning but different phrasing, the collection of these questions is recorded in patterns. The answer is similar in that it holds many responses with the same broad meaning but different terms.



```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": [
        "Hi",
        "Hello",
        "Good day"
      ],
      "responses": [
        "Hey :-)",
        "Hello, thanks for visiting",
      ]
    },
    {
      "tag": "goodbye",
      "patterns": ["Bye", "See you later"],
      "responses": [
        "See you later, thanks for visiting",
        "Have a nice day",
      ]
    },
    …
    {
      "tag": "funny",
      "patterns": [
        "Tell me a joke!",
      ],
      "responses": [
        "Why did the hipster burn his mouth? He drank
the coffee before it was cool.",
      ]
    }
  ]
}
```

## IV. Proposed Method

Feed-Forward Neural Network is a classification algorithm. It is an artificial neural network in which the nodes don't form any cycle or directed acyclic graph. It is the simplest form of neural network. Its opposite is a recurrent neural network in which the nodes form a cycle.

It forms a single-layer perceptron. The structure of a feed-forward is an advantage for certain applications. Examples of the feed-forward neural network are radial basis function networks, which uses activation function.

**How does a Feed-Forward Network work?**

It is a single-layer perceptron. In this model series of input enters the layer and is then multiplied with its weights. All the values are added to get the sum of weighted inputs and check if the value is above a specific threshold. If it is then it is set to zero and if less than the threshold then it is set to -1. This model is mostly used in classification tasks. This process of training and learning produces a form of gradient descent.

Using a property known as the delta rule, the neural network can compare the outputs of its nodes with the intended values, thus allowing the network to adjust its weights through training to produce more accurate output values.

Feed-Forward neural networks have a very simple network structure. Hence to improve the chatbot we can use complex models such as RNN(seq2seq model) or Bidirectional Recurrent Neural Network (BRNN). The BRNN was chosen, like conversation or input to the Chatbot is dynamic, which means the length of the input is unfixed.

## V. Implementation

We tokenized and stemmed the words after loading the JSON file containing roughly 2800 intents and their associated tags, patterns, and answers. Tokenization is the process of extracting an array of distinct words from a phrase, whereas stemming is the process of reducing a word to its word stem, which affixes to suffixes and prefixes or to the roots of words known as lemma. The stem 'tak' will match words like 'take,' 'taking,' 'takers,' and so on. We could tidy up the words list and eliminate any duplicates, but this will be enough for now.

The loaded data will be converted into an array of words, an array of tags, and an array of sets where words with their respective tags will be generated.

```
Tags : ['delivery', 'funny', 'goodbye',
'greeting', 'items', 'payments', 'thanks']

All Words : ["'s", 'a', 'accept', 'anyon', 'are',
...,'what', 'when', 'which', 'with', 'you']

XY : [(['Hi'], 'greeting'), (['Hey'],
'greeting'), (['How', 'are', 'you'], 'greeting'),
..., (['When', 'do', 'I', 'get', 'my',
'delivery', '?'], 'delivery'), (['Tell', 'me',
'a', 'joke', '!'], 'funny'), (['Tell', 'me',
'something', 'funny', '!'], 'funny'), (['Do',
'you', 'know', 'a', 'joke', '?'], 'funny')]
```

Unfortunately, this data structure will not function with PyTorch; we must convert it from documents of words to an array of integers. The bag of words function is used to perform this conversion. The dataset is now ready for training and looks like this:

```
Dataset :

array([[0., 0., 0., ..., 0., 0., 0.], [0., 0., 0.,
..., 0., 0., 0.], [0., 0., 0., ..., 0., 0., 1.],
..., [0., 1., 0., ..., 0., 0., 0.], [0., 0., 0.,
..., 0., 0., 0.], [0., 1., 0., ..., 0., 0., 1.]],
dtype=float32)
```

The dataset presented below is used to load the data into PyTorch, with batch size and shuffle specified in the hyperparameters. The PyTorch-loaded data is processed through the Neural Net Class once more, with the input size, hidden size, and output size supplied as hyperparameters. The size of the input array will equal the size of the training array, and the size of the output array will be the number of tags. As a result, the model can categorize a phrase or an array of words based on their probabilities. The activation function ReLU

(Rectified Linear Unit) is used to turn the model's output into tags with probability.

The Cross-Entropy Loss is assessed during model training, and the Adam Optimizer is applied to the model parameters. The model is then trained for 200 epochs, with each step dumping the model state, input size, hidden size, output size, and all words and tags into a "*.pth" file to be utilized later. For deployment, the file with the lowest Cross-Entropy Loss is utilized.

Before putting it into the model, the user-asked query is tokenized, stemmed, and turned into a bag of words. The model answers with the corresponding tags and their level of confidence. The response is chosen based on the tags with the highest level of confidence.

The model is hosted on Heroku Server and has a REST API. To receive a response, a post request containing a question must be submitted. All of this is served in a Next Js application and is accessible to all Chatbot users (ml-chatbot.vercel.app).

Query :

```
{
   "question":"Hey"
}
```

Response :

```
{
  "answer": "Hi there, how can I help?",
  "isAnswered": true,
  "probability": 0.9999889135360718,
  "probableAnswer": null
 }
```

## VI. Conclusion

Chatbots are programs built to automatically engage with received messages. Chatbots can be programmed to respond the same way each time, to respond differently to messages containing certain keywords and even to use machine learning to adapt their responses to fit the situation. For many applications, the chatbot is connected to the database. The database is utilized to sustain the chatbot and provide appropriate responses to every user. NLP can translate human language into data information with a blend of text and patterns that can be useful to discover applicable responses.

There are NLP applications, programming interfaces, and services that are utilized to develop chatbots. And make it possible for all sorts of businesses – small, medium or large-scale industries. The primary point here is that smart bots can help increase the customer base by enhancing the customer support services, thereby helping to increase sales. AI-powered chatbots understand free language, but also have a predefined flow to make sure they solve the user's problem. They can remember the context of the conversation and the user's preferences.

These chatbots can jump from one point of conversation scenario to another when needed and address random user requests at any moment. These chatbots use Machine Learning, AI and Natural Language Processing (NLP) to understand people.

The goal of NLP is to make the interaction between computers and humans feel like communication between two people. With the help of NLP people can freely interact with chatbots asking a question.

## Acknowledgment

We would like to express our special thanks of gratitude to Athira.M. Nambiar as well as our principal Dr. Shanker P who gave us the opportunity to do this wonderful project on the topic Chatbot using Pytorch, which also helped us in doing a lot of Research and came to know about so many new things. We are really thankful to them.

## References

[1] S., V., & S. (n.d.). Open Data Chatbot.

[2] P., R., ., & A. (n.d.). Feedback-Based Self-Learning In Large-Scale Conversational AI Agents.

[3] Chatbot Tutorial — PyTorch Tutorials 1.10.0+cu102 Documentation. (n.d.). Chatbot Tutorial — PyTorch Tutorials 1.10.0+cu102 documentation. https://pytorch.org/tutorials/beginner/chatbot_tutorial.html.

[4] Chat Bot With PyTorch - NLP And Deep Learning | Python Engineer. (2020, June 14). Chat Bot With PyTorch - NLP And Deep Learning | Python Engineer. https://www.python-engineer.com/posts/chatbot-pytorch/.

[5] Knuth G. (2017, September 9). Contextual Chatbots With Tensorflow | By Gk_| Chatbots Magazine. Medium. https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077.