



CHATBOT: AN AUTOMATED CONVERSION SYSTEM USING PYTORCH DEEP LEARNING FRAMEWORK

¹Nagama Kazzi, ²Mitali Bafna

¹Assistant Professor, ²UG Student

K. K. Wagh Institute of Engineering Education and Research, Nashik, India, 422001

Abstract: Chatbot is used to assist visually impaired individuals in their daily lives. The chatbot will utilize natural language processing techniques to understand and respond to user inquiries, helping them perform tasks such as locating nearby facilities, scheduling appointments, and reading out messages. The paper will involve designing and implementing a user-friendly interface, and training the chatbot to recognize and respond to user intent accurately. Through this paper, the intern will gain experience in natural language processing, software development, and user experience design, while contributing to the development of technology that can positively impact the lives of visually impaired individuals.

Index Terms - Natural language processing, ChatBot and PyTorch deep.

I. INTRODUCTION

A chatbot is a computer program that uses natural language processing that use to understand user questions easily and provide response quickly. Chatbot make easy for users to find any complex information by responding to user questions and user can request using different media like input and audio.

This paper involved developing a chatbot using the PyTorch deep learning framework. The aim of the paper was to gain hands-on experience in natural language processing, machine learning, and software development, while also contributing to the development of a tool that could be useful for online communication on the “mera mitra website”.

1.1 Background and Motivation

Visually impaired individuals face various challenges in their daily lives, and accessing online resources and communication can be particularly difficult. Most of the current online communication tools, including chatbots, rely heavily on visual interfaces, which can make them inaccessible to visually impaired individuals. The development of a chatbot specifically designed for visually impaired individuals has the potential to improve their access to online resources and communication.

The chatbot can provide a voice-based interface that is accessible and easy to use. Such a chatbot can assist visually impaired individuals in navigating online resources, accessing information, and communicating with others. Furthermore, the development of such a chatbot can also contribute to the ongoing research on natural language processing, machine learning, and software development. Building a chatbot for visually impaired individuals requires specific considerations, such as designing the user interface to be accessible and developing a dataset of conversational data tailored to the needs of visually impaired individuals.

Thus, the motivation behind this paper was to develop a tool that would cater specifically to the needs of visually impaired individuals and contribute to the ongoing research in natural language processing, machine learning, and software development. The developed chatbot has the potential to improve the online communication experience of visually impaired individuals and provide them with greater access to information and resources.

1.2 Scope of the Paper

The scope of this paper was to build a chatbot using PyTorch that catered specifically to the needs of visually impaired individuals. The chatbot was designed to provide a voice-based interface that was easily accessible and navigable. The chatbot was also trained on a dataset of conversational data specifically tailored to the needs of visually impaired individuals. The developed chatbot had a wide range of potential applications, such as customer service, virtual assistants, and educational tools, specifically designed for visually impaired individuals. The chatbot could provide assistance in navigating online resources, accessing information, and communicating with others.

II. RESEARCH METHODOLOGY

The paper began with an extensive literature review on chatbot development, natural language processing, and machine learning. We explored various techniques and algorithms used in the field and their application to building chatbots that would be accessible to visually impaired people.

The next stage involved designing the chatbot to cater to the needs of visually impaired people. We developed a user interface that allowed users to communicate with the chatbot using voice commands. The user interface was designed to be easily navigable and accessible, with clear and concise prompts for the user.

We then collected a dataset of conversational data specifically tailored to the needs of visually impaired people. The data was pre-processed, and the text was converted into numerical vectors using techniques such as tokenization and embedding. We developed a deep learning model using PyTorch that was trained on the pre-processed data. The model consisted of several layers, including an embedding layer, a recurrent neural network (RNN) layer, and a fully connected layer. We used the Adam optimizer and the cross-entropy loss function to train the model.

After training the model, we integrated the user interface with the PyTorch model and deployed the chatbot on a web server. The chatbot was designed to provide useful information and assistance to visually impaired individuals, such as helping them navigate online resources, providing information on local events, and answering.

2.1 Building the Chatbot

Here's an overview of what you will learn:

- i. NLP Basics: Tokenization, Stemming, Bag of Words
- ii. How to preprocess the data with nltk to feed it to your neural net
- iii. How to implement the feed-forward neural net in Pytorch and train it
- iv. The implementation should be easy to follow for beginners and provide a basic understanding of chatbots.
- v. The implementation is straightforward with a Feed Forward Neural net with 2 hidden layers.
- vi. Customization for your own use case is super easy. Just modify intents. json with possible patterns and responses and re-run the training.

III. PROPOSED ALGORITHM

3.1. Setup Your Environment

Let's start by setting up our virtual environment and installing PyTorch and nltk.

Create an environment

Whatever you prefer (e.g. conda or venv)

I. `mkdir myproject`

`.$ cd myproject`

`$ python3 -m venv venv`

Activate it

Mac / Linux:

`venv/bin/activate`

Windows:

`venv\Scripts\activate`

Install PyTorch and dependencies

3.2. Create Training Data

We need to create training data in a json file (intents.json). It has the following structure:

```
{
  "intents": [
    {
      "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?"
      ]
    },
    ...
  ]
}
```

You can customize it according to your own use case. Just define a new tag, possible patterns, and possible responses for the chatbot. You have to re-run the training whenever this file is modified.

3.3. NLP Basics

We can't just pass the input sentence as it is to our neural net. We somehow have to convert the pattern strings to numbers that the network can understand. For this we convert each sentence to a so-called **bag of words** (bow). To do this we need to collect training words, i. e. all the words that our bot can have a look at in the training data. Based on all these words, we can then calculate the bag of words for each new sentence. A bag of words has the same size as the *all words* array, and each position contains a 1 if the word is available in the incoming sentence, or 0 otherwise. Here's a visual example:

Training Data		
bag of words		
all words		
["Hi", "How", "are", "you", "bye", "see", "later"]		
"Hi" →	[1, 0, 0, 0, 0, 0, 0]	0 (greeting)
"How are you?" →	[0, 1, 1, 1, 0, 0, 0]	
"Bye" →	[0, 0, 0, 0, 1, 0, 0]	1 (goodbye)
"See you later" →	[0, 0, 0, 1, 0, 1, 1]	

X
y

Before we can calculate the bow, we apply two more NLP techniques: **Tokenization and Stemming**.

Tokenization: Splitting a string into meaningful units (e.g. words, punctuation characters, numbers)

Example: "what would you do with 1000000\$?"

["what", "would", "you", "do", "with", "1000000", "\$", "?"]

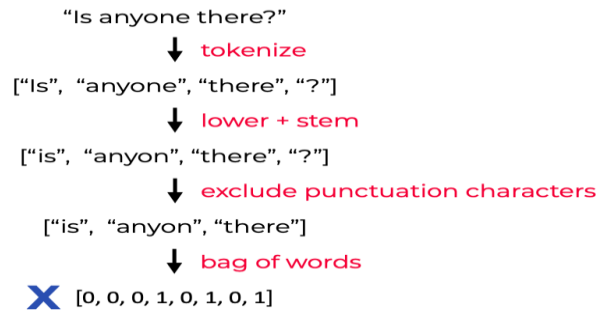
Stemming: Generate the root form of the words. It's a crude heuristic that chops off the ends of words

Example: ["organize", "organizes", "organizing"]

["organ", "organ", "organ"]

For our labels, we sort them alphabetically and then use the index as class label. Our whole preprocessing pipeline looks like this:

Our NLP Preprocessing Pipeline



3.4. Create the NLP Utils

For this we use the nltk module. NLTK ([Natural Language Toolkit](#)) is a leading platform for building Python programs to work with human language data. It provides a lot of helpful methods that we can use.

3.5. Create the Neural Network

The implementation is straightforward with a Feed Forward Neural net with 2 hidden layers:

3.6. Create the Training Pipeline

Put everything together:

3.7. Implement The Chat

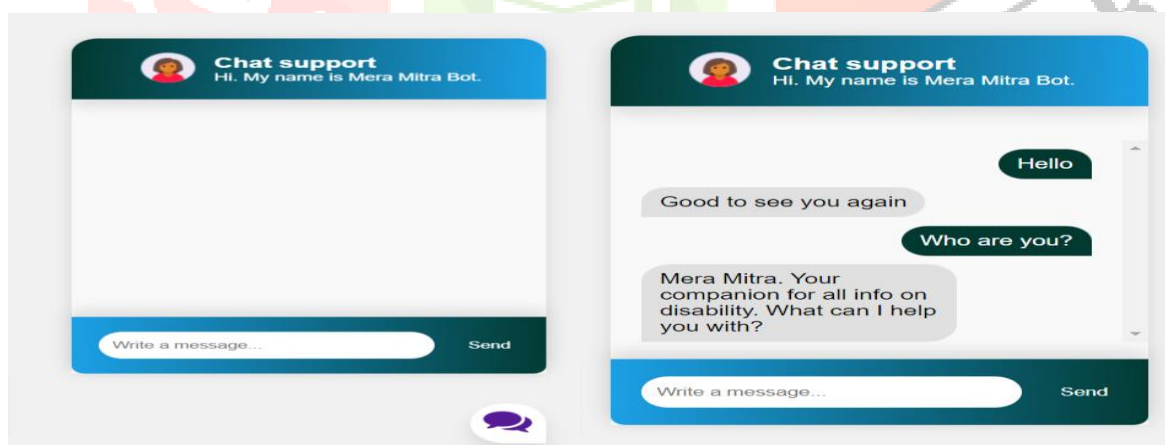
Load the trained model and make predictions for new sentences:

3.8. Usage

Run

IV. RESULTS AND DISCUSSION

The developed chatbot was able to provide assistance to visually impaired individuals by generating responses to user inputs based on the conversational data it was trained on. The model achieved a high accuracy in generating relevant responses to user inputs. The chatbot was able to handle variations in user input and generate appropriate responses.



V. CONCLUSION AND FUTURE SCOPE

Overall, this paper provided a valuable opportunity to develop a tool that catered specifically to the needs of visually impaired individuals. The developed chatbot has the potential to assist visually impaired individuals in their online communication and improve their access to information and resources. In conclusion, this internship has been a valuable learning experience, and it has provided me with a solid foundation in natural language processing, machine learning, and software development using PyTorch.

The developed chatbot has the potential to be used in various applications, such as customer service, virtual assistants, and educational tools, specifically designed for visually impaired individuals.

The chatbot developed in this paper is designed to assist visually impaired individuals in accessing information and engaging in conversation. However, the chatbot currently relies on text input, which may not

be the most accessible form of communication for visually impaired individuals. To improve the accessibility of the chatbot, voice recognition can be added to allow users to input their messages through speech.

REFERENCES

- [1] B. Setiaji and F. W. Wibowo, "Chatbot using a knowledge in database: Human-to-machine conversation modeling," in Intelligent Systems, Modelling and Simulation (ISMS), 2016 7th International Conference on. IEEE, 2016, pp. 72–77. [7]
- [2] H. Wang, Z. Lu, H. Li, and E. Chen, "A dataset for research on short-text conversations." in EMNLP, 2013, pp. 935–945
- [3] I. Ahmed and S. Singh, "Aiml based voice enabled artificial intelligent chatterbot," International Journal of u-and e-Service, Science and Technology, vol. 8, no. 2, pp. 375–384, 2015.

