Generative AI for Impact

Team ASKK

Overview

In the era of AI, ChatGPT just helped people find the answers to queries very instantly; As a student, we still struggle to find answers to many questions related to courses, templates, interns, program changes such as dual degree, double major, etc., and regarding placements.

We have tried to help the Civil Engineering students at IITK by preparing a Customized ChatBot, CivilHelpIITK.

Description of Generative AI model

We have prepared a Customized ChatBot, Its GUI is prepared using Tkinter, a Python library that enables us to develop graphical user interfaces (GUIs). Other Python libraries which are used are numpy, nltk, Keras, and pickle to data cleaning, along with performing linear algebra and the deep learning algorithms.

It will help the Civil engineering students in their various departmental queries.

Architecture

In the folder there are 6 files;

1. intents.json
2. words.pkl
3. classes.pkl
4. chatbot_model.h5
5. train_ChatBot.ipynb
6. Chat.ipynb

- ★ In **train_ChatBot.ipynb** there is a program used to read natural language data into a training set and build a model with a Keras sequential neural network.
- ★ **Words.pkl** and **classes.pkl** both files contain the main words used for recognition.
- ★ intent.json file contains possible datasets.
- ★ **Chatbot_model.h5** contains the actual model created by train ChatBot.ipynb and used by **Chat.ipynb.**
- ★ Chat.ipynb contains the program that cleans up results in accordance with model predictions.

Libraries used are numpy, nltk, Keras, and pickle which are used for cleaning up text, preparing it for deep learning algorithms, loading json files directly into Python, performing linear algebra operations, for using deep learning framework.
To train the model, we will employ stochastic gradient descent (SGD), which may seem intricate. However, it's worth noting that SGD is a more efficient alternative to standard gradient descent.
Upon completing the model training, the entire structure will be converted into a numpy array and saved as chatbot_model.h5.
Using this trained model, we can construct the interface for our chatbot.
Our GUI encompasses several functions that encapsulate essential processes. These functions are designed to work together seamlessly. Let's explore them:

1. clean_up_sentence(): This function receives an input sentence and performs necessary cleaning operations. It prepares the sentence for further processing. This function is utilized within the bow() function.

2. bow(): Using the cleaned sentences, this function generates a bag of words. The bag of words is crucial for predicting classes based on the training results obtained earlier. It essentially represents the words present in the input sentences.

3. predict_class(): Within this function, an error threshold of 0.25 is employed to prevent excessive overfitting. The purpose of this function is to provide a list of intents along with their corresponding probabilities. These probabilities indicate the likelihood of a given intent being a suitable match.

4. getResponse(): Taking the list of intents and their probabilities as input, this function checks a JSON file containing predefined responses. It selects the most appropriate response based on the highest probability.

5. chatbot_response(): This function acts as the core component of our chatbot. It takes a message as input (which is provided through the chatbot GUI). It utilizes the predict_class() function to predict the class of the message, obtains the resulting list

of intents and probabilities, and passes them to the getResponse() function. The final output is the chatbot's response to the user input.

Together, these functions form the foundation of our chatbot. They allow us to interact with the bot by inputting messages and receiving corresponding responses.

Training Parameters

As it is a customized bot, we don't have the training dataset available, we have manually created most of the data.

The dataset can be found in the intent.json file. Our json file was extremely tiny in terms of the variety of possible intents and responses.

Performance metrics

We have very limited time and not pre-available dataset, and due to manually added dataset, the accuracy of model is not too much, but once we get enough time to work on the model we can prepare plenty of dataset and can take help of survey forms and can add the website data, and various data regarding internship and placement.

Working Principle and demo of model

In the code first, we take the words list lemmatized and lowercase all the words for not having so much confusion. After sorting the lists we built a deep learning model in which we initialized our training data with variable training. We have a feature named output_row that merely serves as a list's key. Then, using a train-test-split with the patterns as the X variable and the intents as the Y variable, we randomize our training set.
Once our training and test data is prepared, we can proceed to utilize the **Sequential model** from Keras for deep learning. The Sequential model serves as a straightforward neural network known as a multi-layer perceptron.

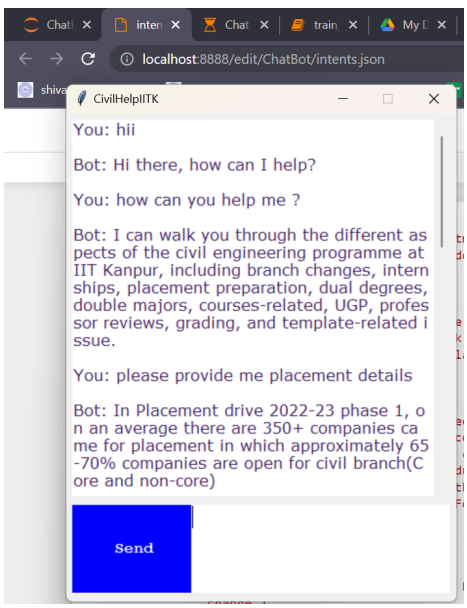
This particular Sequential model consists of three layers. The first layer comprises 128 neurons, followed by a second layer with 64 neurons. The third layer contains a number of neurons equivalent to the total intents we aim to predict.

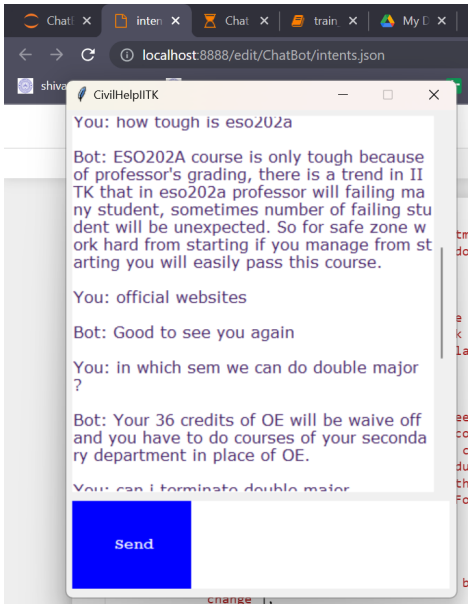
Firstly, we define a function called send(), which establishes the fundamental functionality of our chatbot. When we input a message into the chatbot, it will generate a response using the chatbot_response() function, provided that the message is not an empty string.

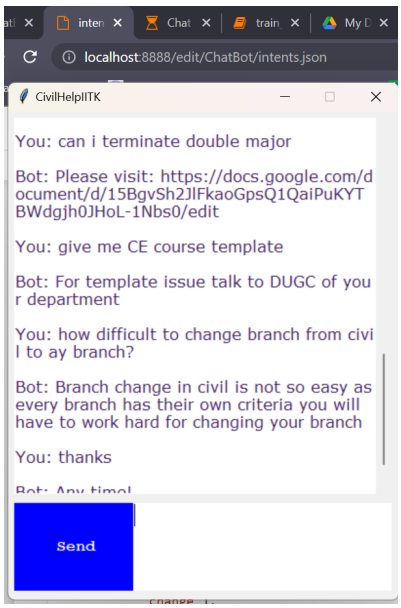
Next, we proceed to construct our chat window, scrollbar, message-sending button, and textbox, utilizing the Tkinter library. We position these components on the screen using simple coordinates and heights.

By organizing the elements in this manner, we are able to create an engaging and interactive GUI for our chatbot.
We run our chatbot(CivilHelpIITK) on a jupyter notebook.

Here is some images how our chatbot is responding:







Here are some examples of question we can ask:

Q. How many companies came for civil
Q. Average CTC for civil branch in placement
Q. Good companies open for Civil non-core
Q. Please provide me details of this course.
Q. Give me CE course template
Q. Can I terminate dual degree anytime
Q. How difficult to change branch from civil to any branch?
Q. How to get a good project?
Q. Is there any prerequisite for getting project ?
Q. How to take Industrial Intern in civil?
Q. CPI Criteria for core internship

There are a lot of questions we can ask, but occasionally the ChatBot may give the incorrect response due to its lack of accuracy. Additionally, due to time restrictions, we are unable to create a large dataset, so we put some basic questions first. If this project is chosen for further high-level consideration, it may be a good project.

Submitted By

Ayushi Agrawal	200257
Khushbu	200512
Koustav Saha	200520
Shivangi	200942