DGIBOT: Chat Assistant for Institution Queries

Sameer Aggarwal[1], Pranjal Bhardwaj[2]

[1,2]Departement of CSIT, Dronacharya Group of Institutions, Greater Noida, India

Dr. Bipin Pandey

Department of CSIT, Dronacharya Group of Institutions, Greater Noida, India

Abstract - The fast advancements in artificial intelligence and natural language processing have led to a new era of conversational agents known as chatbots. There is a lot of interest in these advanced virtual assistants coming from many other sectors, including education. The study aims to develop an API and chatbot functionality for college students. A chatbot that uses Deep Learning (DL) and Natural Language Processing (NLP) to answer common inquiries about the Dronacharya Group of Institutions is being developed to improve user interaction and efficiency. The chat bot programme, which can communicate with users in a human-like manner, assists students, parents, teachers, and applicants for admission by providing information about the available courses, professors, facilities, locations, and bus routes. Etc. Almost every query is responded to using the model, and any that cannot be is recorded in a database for processing and bot improvement. By maximising its potential, this study explores how chatbot technology has an impact on administrative processes, the student experience, and overall institutional effectiveness.

Keywords – Artificial Intelligence (AI), Natural Language Processing (NLP), deep learning (DL), Query, chatbot, Web APIs, educational Institutions

I. INTRODUCTION

A Chatbot can be described as software or Script that has the capability to chat with people on behalf of an organization, group or individual to provide answers related to some general commonly known questions or Frequently Asked Questions (FAQs). This is accomplished by creating a mapping between the questions and their responses. Or, if we want to take the advantage of the latest technologies, using Artificial Intelligence.

This software is used to perform tasks such as quickly responding to users, providing them information, help them in making and correct and quick decision on products or service and providing better service to customers.

In this study, we present the general working principle and the basic concepts of our Artificial Intelligence based chatbot for our college Dronacharya Group of Institutions. By performing this study, we hope to highlight the revolutionary potential of chatbot technology within educational institutions and offer insightful contributions to the academic community. The results of this study will enable decision-makers at the Dronacharay Group of Institutions and other comparable organisations better understand the advantages and difficulties of implementing chatbots, which will eventually improve the academic experience for all parties concerned.

The work structure of this paper is as follows section II includes brief overview of similar works that are being present. In section III methodology of the document are described, followed by proposed work and experimental result and analysis in section IV and V. The paper ends with the conclusion of the experiments in section VI and section VII covers all the references of proposed work.

II. LITERATURE SURVEY

An end-to-end memory network for question-answering tasks is proposed by Sukhbaatar et al. [1] and can be expanded to create chatbots. The conversational history is stored in the model using a memory component, and the necessary information is selectively retrieved from the memory using attentional mechanisms.

Using an encoder-decoder framework, Vinyals and Le [2] describe a sequence-to-sequence neural network model for producing human-like responses in a conversation. The model has been demonstrated to produce coherent and contextually appropriate responses after being trained on a sizable corpus of conversational data.

The Transformer design, introduced by Vaswani et al. [3], has been used to develop chatbots. Because the model is entirely based on self-attention mechanisms, it is highly parallelizable and efficient. In a range of natural language processing tasks, transformers outperform classic sequence-to-sequence models.

BERT, a pre-trained language model that has been fine-tuned for a range of NLP applications, including chatbot development, is introduced by Devlin et al. [4]. BERT is a bidirectional model that has been pre-trained on a huge corpus of text and has been shown in many NLP benchmarks to exceed earlier state-of-the-art models.

Gu et al. [5] develop a conditional Wasserstein auto-encoder for producing multimodal responses in a discussion, which can be utilised to create more engaging chatbots. The model has been trained to produce responses that are not only consistent and relevant, but also varied and interesting.

According to Wolf et al. [6], a transfer learning strategy for creating chatbots involves fine-tuning a previously trained neural network using a fresh domain-specific dataset. On several conversational datasets, the model is demonstrated to produce cutting-edge results and can be trained using a small amount of labelled data.

Overall, the studies mentioned above emphasize the relevance of using deep learning approaches while developing chatbots. These models have been demonstrated to produce more coherent and contextually relevant responses while also handling complex conversations. The inclusion of pre-trained models, transfer learning, and multimodal techniques has enhanced these models' performance even further. However, more research is needed to improve the naturalness and diversity of chatbot responses.

Objective of this study is to establish a means of communication that is reliable, fast, data transferred is accurate and consistent. We want to establish a communication mode that colleges can use to provide crucial and correct information to the admission seeking students without being scammed, wasting time, and having to rely on unreliable websites and old information. This report will also examine the web APIs of two prominent chat programmes, Telegram and Discord. The issue is that, while various chat apps have comparable functionality, their web APIs differ in terms of capabilities, performance, and usability. Reviewing technical documentation, testing API endpoints, and analysing code samples are some of the data collection approaches.

III. METHODOLOGY

A. Dataset Description

We created the 'intents.json' file that has a size of 69KB and contains 1234 lines divided in multiple sections. Each different section contains tags and responses related to queries about Dronacharya Group of Institutions. It contains 51 Tags, 657 questions. From every tag, there are approx. 10 queries or questions and 2 to 5 different responses. When the AI model determines the tag of the query/question, the bot choses one responses at random. It will not work unless the response probability is more that the defined error threshold.

B. Technology Stack

1) Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems.

This project is made solely using python i.e., both the backend and frontend of this assistant are coded using python only.

Some of the libraries and modules of python that we are used are:

a) Tflearn: TFLearn is a user-friendly deep learning library built on TensorFlow, offering a simplified API for model development and training. With pre-built layers and models, it enables quick prototyping and customization. TFLearn integrates with TensorFlow's ecosystem, provides visualization tools, and supports data preprocessing, making it a popular choice among deep learning practitioners.

b) Tensorflow: TensorFlow is an open-source deep learning framework created by Google that offers a comprehensive ecosystem for constructing and training machine learning models. It boasts a flexible architecture, enabling the definition and training of diverse neural network architectures. With support for distributed computing, GPU acceleration, and deployment across multiple platforms, TensorFlow has gained significant popularity in both research and industry domains.

c) Nltk: NLTK (Natural Language Toolkit) is a powerful open-source library for natural language processing in Python. It provides a wide range of tools and resources for tasks like tokenization, stemming, tagging, parsing, and sentiment analysis. With its extensive collection of corpora and language models, NLTK aids in building applications that process and analyze textual data efficiently and effectively. It has gained popularity among researchers, educators, and developers for its rich set of functionalities and ease of use.

d) Keras: keras is a user-friendly high-level API in TensorFlow, simplifying deep learning model creation. It supports popular architectures like CNNs and RNNs, abstracting low-level implementation details. Integrated with TensorFlow, it provides GPU acceleration and distributed training.keras is a preferred choice for deep learning practitioners due to its simplicity and flexibility.

e) Logging: Python logging libraries facilitate efficient logging of messages and events in Python applications. They offer features such as log levels, formatting, handlers, and filters. Popular libraries like logging, loguru, and structlog provide flexibility, customization, and easy integration, helping developers in debugging, monitoring, and troubleshooting their code effectively.

f) Requests: The request library in Python provides a simple way to handle HTTP requests. It offers functionalities for sending and receiving data from web APIs, handling various request methods, managing headers and parameters, and handling cookies and sessions. With its user-friendly interface, the request library is widely used for accessing and interacting with web services in Python applications.

g) Pickle: pickle module enables object serialization and deserialization. It allows for easy conversion of complex data structures, such as lists and dictionaries, into a binary format. Pickle is commonly used for storing and retrieving data or transferring objects between different Python programs.

h) urllib3: It is a powerful HTTP library for Python that provides a higher-level interface for making HTTP requests. It offers features like connection pooling, SSL support, and automatic retries, making it a reliable choice for handling HTTP interactions in Python applications.

i) Pandas: Pandas is a versatile data manipulation and analysis library in Python. It offers powerful data structures, like data frames and series, along with a wide range of functions for cleaning, transforming, and analyzing structured data. Pandas is widely used in data science and data analysis workflows for its efficiency and ease of use.

j) SQLite3: SQLite3 is a lightweight and serverless relational database management system (RDBMS) integrated with Python. It offers a simple and efficient solution for storing, querying, and managing structured data in a local database file. SQLite3 is commonly used for embedded systems, small-scale applications, and prototyping due to its ease of use, portability, and zero-configuration setup. It supports standard SQL syntax, provides transactional integrity, data integrity, and concurrent access control. The code for SQLite is in the public domain and is thus free for use.

2) Natural Language Processing

Natural Language Processing (NLP) is the study of letting computers understand human languages. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers do not recognize the words and do not understand the grammars. NLP can be regard as a “translator”, who will translate human languages to computer understandable information.

Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. A single replace of one character or even a space can have significant difference. However, the emergence of NLP is changing the way of interacting. Apple Siri and Microsoft Cortana have made it possible to give command in everyday languages and is changing the way of interacting

3) Deep Learning

Deep learning is a subset of machine learning that is essentially a three- or more-layered neural network. These neural networks seek to imitate the behaviour of the human brain, albeit with limited success, allowing it to "learn" from enormous volumes of data. While a single-layer neural network can still produce approximate predictions, additional hidden layers can assist optimise and improve for accuracy.

Many artificial intelligence (AI) applications and services rely on deep learning to improve automation by performing analytical and physical tasks without human intervention. Deep learning technology is at the heart of both common products and services (such as digital assistants, voice-enabled TV remote controls, and credit card fraud detection) and emerging technologies (such as self-driving cars).

C. API’s Used

1) Telegram

A telegram script is used that handles questions/queries from telegram. The standard gets Updates APIS provided by the telegram are used to fetch user messages. These messages are then passed the model function that generates an response using the model. The final response generated by the model is returned to the telegram which is then cleaned up, formatted and is sent back to the user using the response sendMessage API.

2) Discord

Discord provides a python package that can be installed to create bots that can interact with students on behalf of college. It is a very developer friendly package that can create basic bots in minimum duration of time. This package is used to integrate the AIDL model to interact with students. These bots can chat as private chats or can be invited to guilds. The message send in private chat is directly accepted by the bot as a query. But the messages in guilds need to be prefixed by a special character called “command prefix” to differentiate the messages meant for bots from others. The message received from private chat or guild text channels is processed by the model to generate the response which is then send to the source. Links and files need special handling that is provided by embed module.

3) Flask

Flask is a python-based website development package that provides a very light module for development of web applications. As all the other parts of the DGI-Chatbot are developed with python. It is necessary to create an interface in Python that can communicate with the AI model as well as

HTML code for the website. Flask is a very light weight module that takes minimal work to get a simple website operating required to test the bot's operation across the websites. APIS functionality is provided by Flask. These APIs serve as a link between the model and the website. They oversee receiving user questions and delivering the bot's responses to the user via the internet. Web integration is crucial as almost everyone has access to a web browser, if not to other platforms. This making the chatbot far reached and providing solution to larger group of students very easily.

D. Database Description

For this study we use SQLite database, as it is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The defined database, includes a table called 'messages' in the database.

The 'messages' table consists of two attributes:

- 'id': It serves as the primary key, of type INTEGER, uniquely identifying each row in the table.
- 'message': This attribute, of type VARCHAR with a maximum length of 50 characters, stores textual messages associated with the Telegram application.

Initially we would be saving every query made to the model to check the functionality of the model and working of the database related functions. After the initial phase, we would move to storing only the queries for which the accuracy or the error threshold (set threshold 0.39) is in certain ranges. This would allow as to improve the model and increase overall accuracy. In final phase, we would be analysing the user response to model answer and store the queries for which user response was not satisfactory. Table 1 shows the frequently asked user queries from chatbot which we stored in 'messages' table.

ID	Message
1	cse hod
2	Tell me about college admission
3	course
4	hi
5	events and activities
6	quit()
7	courses offered
8	how are you
9	who is sameer
10	college
11	/start
12	How are you
13	Fee structure
14	/start
15	hello
16	/start
17	fee
18	fees
19	bus routes

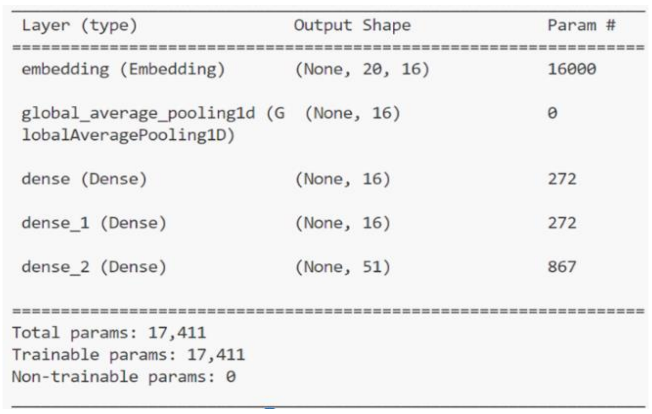
Table -1: Database Entries Sample

E. Models used

In this study we proposed two models built with two different API's one with Keras names 'Keras model' and another is TFLearn based model named 'TFLearn model'. Both, the Keras and TFLearn models are trained on the above-mentioned intent file where dependent variable, train_y is the response, and independent variable, train_x contains tags and patterns from the proposed intent file.

1) Keras Model

The model is built using Keras, which uses Tensorflow in backend by default. The model is a four-layered sequential neural network. The first layer is an Embedding layer, which accepts as arguments the vocabulary size, embedding dimension, and input length. The input length is the longest text sequence in the input. The second layer is a GlobalAveragePooling1D layer that uses global average pooling over the sequence dimension. Dense layers with 16 units each and ReLU activation make up the third and fourth layers. The final layer is a Dense layer with softmax activation and the number of classes as units. The sparse categorical cross-entropy loss function, Adam optimizer, and accuracy metric are used to build the model. Figure 1 shows the model summary.



```
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 20, 16)            16000

 global_average_pooling1d (G  (None, 16)               0
 lobalAveragePooling1D)

 dense (Dense)               (None, 16)                272

 dense_1 (Dense)             (None, 16)                272

 dense_2 (Dense)             (None, 51)                867

=================================================================
Total params: 17,411
Trainable params: 17,411
Non-trainable params: 0
```

Fig -1: Keras Model

2) TFLearn Model

The TFLearn library was used to create this model. The neural network includes an input layer with the same number of neurons as the number of features in the input data, two hidden layers with eight neurons each, and an output layer with the same number of neurons as the number of classes in the output. Because no activation function is specified for the hidden layers, the default activation function (ReLU) is used. The SoftMax activation function is used by the output layer to construct probability distributions over the output classes.

The categorical cross-entropy loss function and the Adam optimizer are used to train the model. The tensorboard_dir argument specifies the location of the TensorBoard logs for visualisation and monitoring of the model training process. This code constructs a simple deep neural network for multi-class categorization. However, because the precise dataset and performance indicators are missing, no judgements regarding the model's performance can be drawn.

F. Hyperparameters Taken

For training Keras model, the hyper parameters are taken that shown in Table 2 and for TFLearn model, first a bag of words is created from cleaned up sentences so an integer array can be created. This model is trained with hyper parameters which are shown in Table 3.

Table -2: Keras Model Hyperparameters

Parameters	**Values**
Output layer activation function	softmax
Number of hidden neurons	16
loss	sparse_categorical_crossentropy
optimizer	adam
metrics	accuracy
epochs	1000
Hidden layers activation function	relu
Vocabulary size	1000
Embedding dimension	16
Oov token	<OOV>

Table -3: TFLearn Model Hyperparameters

Parameters	**Values**
Output activation function	softmax
optimizer	adam
Batch size	8
epochs	1000
Number of hidden neurons	8

IV. PROPOSED WORK

In this study, we created an interactive chatbot for frequently asked questions (FAQs) about our college. The proposed work flow is presented in Figure 2. The Process starts with user sending a message to the bot. As soon as the message is received, the server sends the message data to the bot. Then Bot reads and filters the received data on different parameters and then send the user message to the AI model.

Model converts the natural language message into tokens which then are used to predict and appropriate answer for the message. Answer accuracy is then checked, and if the accuracy is of desired level, the user is prompted with an answer for their query. If the answer accuracy if lower than a set threshold, the message is deemed new. All new messages are added to the database so that they can be used later when it is time to update the Model.

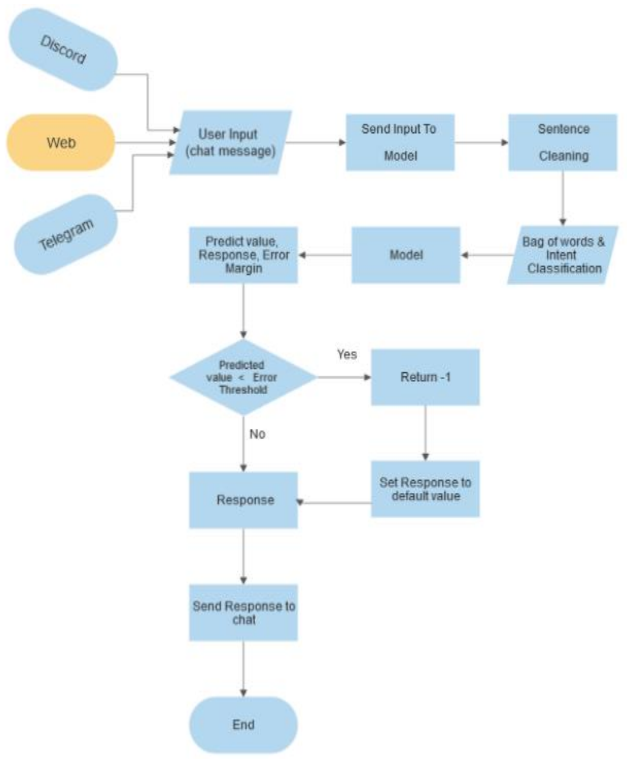
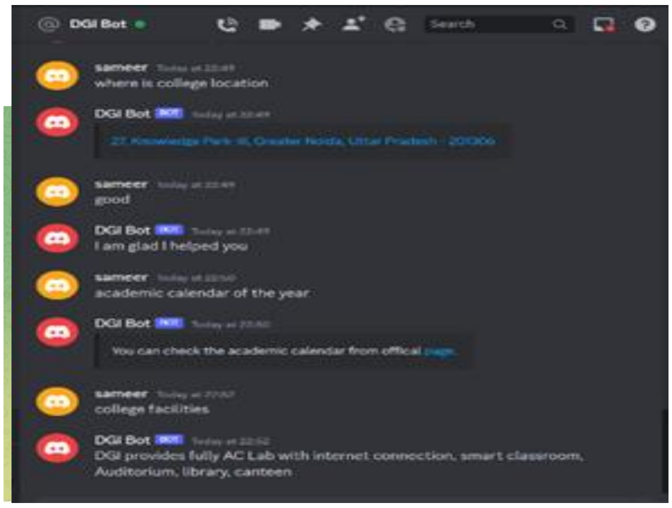

Discord
Web
Telegram
User Input (chat message)
Send Input To Model
Sentence Cleaning
Bag of words & Intent Classification
Model
Predict value, Response, Error Margin
Predicted value < Error Threshold
Yes
Return -1
No
Set Response to default value
Response
Send Response to chat
End

Fig -2: Project Workflow

V. RESULTS

These results are obtained by using the best from both the proposed models. All the Figures mentioned below are the result of running and executing the best model on all three platforms. Results from both the models were obtained by asking a variety of questions to test the models on all parameters including hyperlinks and files. These results were then compared on response time and accuracy of the response and the best model was chosen which was then used to generate the output.

Also, findings of this study indicate that Telegram's web API is more suitable for developers who require a high degree of customization and control over their chat application. Its API offers features such as bots, inline mode, and channels, which enable developers to create complex chat applications with advanced functionality. Discord's web API, on the other hand, is designed for simplicity and ease of use. Its API offers


DGI Bot

sameer
where is college location

DGI Bot BOT
27, Knowledge Park-III, Greater Noida, Uttar Pradesh - 201306

sameer
good

DGI Bot BOT
I am glad I helped you

sameer
academic calendar of the year

DGI Bot BOT
You can check the academic calendar from offical page.

sameer
college facilities

DGI Bot BOT
DGI provides fully AC Lab with internet connection, smart classroom, Auditorium, library, canteen

features such as webhooks, rich embeds, and slash commands, which enable developers to quickly create chat applications with basic functionality.

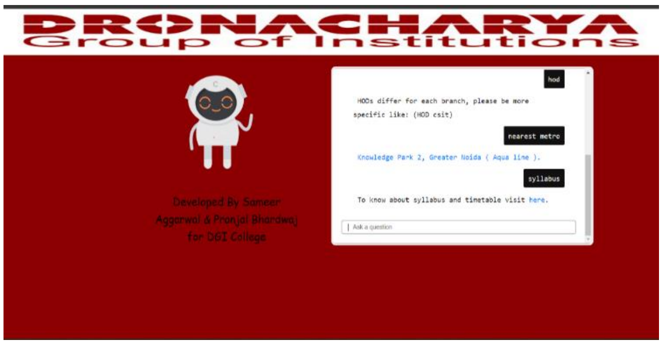
Fig -3: Chatbot response Telegram Snapshot

Fig -4: Chatbot response Web Snapshot

Fig -5: Chatbot Response Discord Snapshot

VI. CONCLUSION

This study is done so that every visitor on specified platform can easily and interactively find all the information related to college Dronacharya Group of Institution. DGI-Chatbot has proven to be a valuable resource for both students and faculty. As it can answer a wide range of queries about college life by utilising natural language processing techniques and machine


DRONACHARYA
Group of Institutions

Developed By Sameer Aggarwal & Pranjal Bhardwaj for DGI College

hod

HODs differ for each branch, please be more specific like: (HOD csit)

nearest metro

Knowledge Park 2, Greater Noida (Aqua line).

syllabus

To know about syllabus and timetable visit here.

Ask a question

learning algorithms. It is also connected with popular messaging systems such as Discord and Telegram, allowing students to access it across several platforms. Overall, the chatbot has proven to be a tremendous tool for FAQs, offering students and employees a simple and accessible way to get rapid and accurate responses to their inquiries. The chatbot's capacity to interface with numerous platforms makes it more accessible to a wider audience, making it even more valuable.

Future study could focus on improving chatbots' natural language processing capabilities for educational institutions, evaluating the impact of chatbots on student engagement and learning outcomes, and investigating the use of chatbots in various industries.

VII. REFERENCES

[1] Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus. "End-to-end memory networks." Advances in neural information processing systems 28 (2015).

[2] Vinyals, Oriol, and Quoc Le. "A neural conversational model." arXiv preprint arXiv:1506.05869 (2015).

[3] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[4] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[5] Gu, Xiaodong, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. "Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder." arXiv preprint arXiv:1805.12352 (2018).

[6] Wolf, Thomas, Victor Sanh, Julien Chaumond, and Clement Delangue. "Transfertransfo: A transfer learning approach for neural network based conversational agents." arXiv preprint arXiv:1901.08149 (2019).