E-BOT: ADVANCED DEEP LEARNING BASED CHATBOT BUILT USING NLP AND KNN

A PROJECT REPORT

Submitted by

Aravind S (2020113002)

Aravinth S (2020113003)

Palaniappan M (2020113306)

In the partial fulfillment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND BUSINESS SYSTEMS

SETHU INSTITUTE OF TECHNOLOGY

(An Autonomous Institution | Accredited with 'A++' Grade by NAAC)

PULLOOR, KARAIPATTI – 626 115

ANNA UNIVERSITY – CHENNAI 600 025

APRIL 2024

SETHU INSTITUTE OF TECHNOLOGY

(An Autonomous Institution | Accredited with 'A++' Grade by NAAC)

PULLOOR, KARAIPATTI – 626 115

ANNA UNIVERSITY – CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report entitled "**E-BOT, Advanced Deep Learning based Chatbot built using Natural Language Processing and Keras Neural Networking**" is the bonafide work of "Aravinth S", "Aravind S" and "Palaniappan M" who carried out the project work under my supervision.

SIGNAURE	**SIGNATURE**
Dr. V. Shunmughavel, M.E., Ph.D.	**Mr. V. Rajesh Kumar M.E., (Ph.D.)**
HEAD OF THE DEPARTMENT	**SUPERVISOR**

Submitted for the 19UCB801 – Project Work, End Semester Examination, held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINE**

ACKNOWLEDGMENTS

First, we would like to thank **GOD** the Almighty for giving us the talent and the opportunity to complete our project and our B. Tech degree.

We wish to express our earned gratefulness to our Honorable Founder and Chairman, **Mr. S. Mohamed Jaleel B.Sc., B.L.** for his encouragement extended us to undertake this project work.

We wish to thank and express our gratitude to our Chief Executive Officer **Mr. S. M. Seeni Mohaideen B.Com., M.B.A.** Director - Administration, **Mrs. S. M. Nilofer Fathima B.E., M.B.A.,** Director - R&D, **Dr. S. M. Nazia Fathima B.Tech., M.E., Ph.D.** and Joint Chief Executive Officer, **Mr. S. M. Seeni Mohamed Aliar Maraikkayar B.E., M.E., M.B.A., (Ph.D.)** for their support in this project.

We would like to thank and express our gratitude to our Principal, **Dr. G. D. Siva Kumar B.E., M.E., Ph.D.,** for providing all necessary for the completion of the project.

We wish to express our profound gratitude to our Head of the Department **Dr. V. Shunmughavel, B.E., M.E., Ph.D.,** for granting us the necessary permission to proceed with our project.

We are immensely grateful to our guide, supervisor, and overseer **Mr. V. Rajesh Kumar M.E., (Ph.D.)** for encouraging us a lot throughout the course of the project. We render our sincere thanks of his support in completing this project successfully.

We thank our **Parents, Faculty Members, Supporting Staff and Friends** for their help extended during these times.

ABSTRACT

E – BOT is a chatbot tailored to assist users navigating *Tamil Nadu E-Service websites*, employing Advanced *Natural Language Processing* (NLP) and Deep Learning technologies. The primary goal is to develop a conversational agent capable of understanding user queries related to e-services and providing relevant assistance. At its core, the project harnesses the *Natural Language Toolkit* (NLTK) in Python for effective language processing. NLTK aids in breaking down and understanding words, forming the basis for the chatbot's ability to comprehend user input. By utilizing NLTK, the project aims to enhance the chatbot's language understanding, ensuring it can effectively interpret and respond to user questions regarding Tamil Nadu E-Service portals. The architecture of the chatbot relies on a neural network model constructed using the Keras framework. Trained on a dataset encompassing various intents and patterns related to e-services, the model becomes adept at recognizing and categorizing user input accurately. Deep learning techniques embedded in the model enable it to adapt and improve its performance over time. The training process involves creating a bag-of-words representation for each input pattern. Configured with multiple layers, including densely connected layers and dropout layers, the model optimizes its ability to understand patterns from the training data. For user interaction, the chatbot is deployed using the Flask framework, transforming it into a web-based application. This user-friendly interface allows individuals to input queries, receive responses, and engage in dynamic conversations with the chatbot regarding Tamil Nadu E-Services. The project outcome is an intelligent chatbot capable of understanding and addressing user queries related to Tamil Nadu E-Services. The integration of NLP and deep learning technologies contributes to the creation of a chatbot that continually evolves and refines its language understanding skills.

In addition to its technical functionalities, the project pays keen attention to the user experience and practical utility of the chatbot within the Tamil Nadu E-Service ecosystem. The web-based interface provides a simple and intuitive platform for users to seamlessly interact with the chatbot, emphasizing functionality and ease of use over unnecessary complexities. In conclusion, this project represents a Spartan yet impactful approach to the development of a chatbot, specifically tailored for assisting users with Tamil Nadu E-Services. By integrating Natural Language Processing and Deep Learning technologies, the chatbot not only signifies a technical achievement but also addresses practical user needs within the context of E-Governance. The synergy of simplicity and effectiveness showcased in this project lays the groundwork for an advanced conversational agent, contributing to a more accessible and user-friendly experience for individuals navigating Tamil Nadu E-Service websites.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

ABBREVIATIONS	EXPANSIONS
AI	Artificial Intelligence
ML	Machine Language
DL	Deep Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
GPU	Graphical Processing Unit
CPU	Central Processing Unit
RAM	Random Processing Unit
SSD	Solid State Drive
HDD	Hard Disk Drive
API	Application Programming Interface
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
JSON	Java Script Object Notation
AIML	Artificial Intelligence Markup Language
URL	Uniform Resource Locator
FAQ	Frequently Asked Questions

CHAPTER – 1

INTRODUCTION

1.1. Overview of the Project

The project aims to deploy an artificial intelligence-driven chatbot system across all Tamil Nadu government websites. This initiative seeks to revolutionize the user experience by introducing a fresh approach to accessing online government services. By utilizing cutting-edge technologies like machine learning, natural language processing, and deep learning, the chatbot will act as a virtual assistant, guiding users through website navigation, assisting with transactions, and providing timely information. By leveraging these advanced technologies, the chatbot will empower users to navigate government websites effortlessly, simplifying complex processes and ensuring easy access to essential services. The integration of the chatbot into existing infrastructure is expected to streamline processes, enhance user satisfaction, and improve overall efficiency in service delivery [1]. Ultimately, the project aims to drive the digital transformation of public services in Tamil Nadu, making them more accessible, efficient, and responsive to citizens' needs. Through innovative initiatives, the project seeks to empower citizens, improve service delivery, and create a more inclusive society.

1.2. Motivation of the Problem

The project's motivation stems from the pressing need to address the challenges faced by users in accessing and navigating government websites efficiently. With an ever-expanding digital landscape and increasing reliance on online platforms, users often encounter complexities and barriers that hinder their ability to access vital information and services. These challenges lead to frustration, inefficiencies, and reduced engagement with government websites. Recognizing the need for innovative solutions, the project seeks to harness the power of artificial intelligence and chatbot technology to

overcome these obstacles, providing users with a more intuitive and user-friendly experience.

1.3. Objective of the Project

The primary objective of the project is to deploy an advanced chatbot system on Tamil Nadu government websites, with the overarching goal of enhancing user experience and efficiency. By leveraging cutting-edge technologies such as artificial intelligence, machine learning, and natural language processing, the chatbot will be capable of understanding user queries, providing relevant information, assisting with transactions, and facilitating seamless navigation [2]. Through these capabilities, the project aims to optimize the functionality of government websites, improve citizen engagement, and promote digital inclusivity across Tamil Nadu. Additionally, the project aims to set a precedent for digital innovation and modernization within the government sector, paving the way for future advancements in public service delivery and citizen interaction.

1.4. Usefulness / Relevance to the Society

The project holds immense usefulness and relevance to society by addressing critical challenges faced by users in accessing government services and information online. By deploying a chatbot, the project aims to access to essential resources, streamline processes, and improve overall efficiency in service delivery. This initiative not only enhances user experience but also fosters greater transparency, accountability, and accessibility. Furthermore, by leveraging cutting-edge technologies and innovative solutions, the project sets a precedent for digital transformation and modernization, driving socio-economic development and empowering citizens across Tamil Nadu. Through these efforts, the project aims to create a more inclusive, efficient, and user-centric digital ecosystem, ultimately benefiting society.

CHAPTER – 2

LITERATURE SURVEY

The literature survey encompasses a comprehensive exploration of various types of chatbots and their functionalities, as well as related research and developments in the field.

2.1. Types of Chatbots

2.1.1. Menu / Button Based Chatbots

These chatbots offer a straightforward user experience with pre-defined answers presented in the form of buttons. Operating on decision tree principles, they provide consistent responses to user queries.

2.1.2. Keyword Recognition - Based Chatbot

This type of chatbot relies on specific keywords within user queries to generate responses. However, limitations arise from the potential failure to process entire inputs and redundancy issues [3].

2.1.3. Contextual Based Chatbot

Considered the most advanced, these chatbots utilize machine learning and artificial intelligence technologies to understand user queries and improve efficiency over time [4][5].

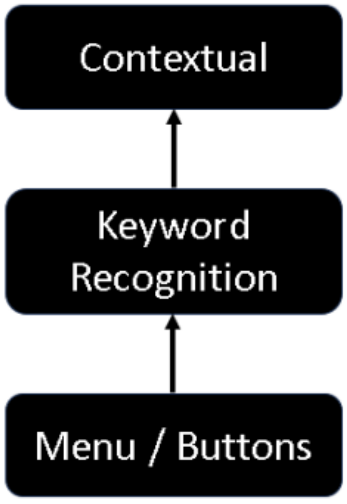

FIG 2.1 PREFERENCES OF CHATBOT

2.2. Related Works

2.2.1. A.L.I.C.E. Chatbot System

The Alice Chatbot System, implemented by Wallace in 1995, uses AIML files to store conversational patterns. AIML consists of categories and topics, with categories serving as rules for generating responses based on user inputs [5].

2.2.2. Dedicated Chatbots for Websites

Increasingly popular for providing real-time customer support, dedicated chatbots cater exclusively to individual websites, offering round-the-clock assistance to visitors [6].

2.2.3. Advanced AI – Based Chatbots

Utilizing techniques like machine learning and natural language processing, these chatbots analyze user inputs and improve accuracy over time. Libraries like Chatterbot facilitate the development of AI-based chatbots by providing tools for training and processing natural language inputs [3][7].

CHAPTER – 3

DESIGN

3.1. System Architecture

The system architecture of E-Bot encompasses the design and organization of its various components to ensure efficient performance and scalability. The architecture is designed to handle user queries, process data, and interact with government websites seamlessly. Key components of the system architecture include:

Client Interface: The user interacts with E-Bot through a web-based interface, providing queries and receiving responses. HTML, CSS, and JavaScript are fundamental technologies used for building interactive and visually appealing web applications. HTML provides the structure of the web page, CSS adds styling and layout, while JavaScript enables dynamic behavior and interactivity, enhancing the user experience. Together, these

technologies form the backbone of modern web development, enabling the creation of responsive and feature-rich client-side interfaces.

Server-Side Processing: Upon receiving a query, the server processes the request, utilizes the chatbot's algorithms to understand the query, fetches relevant information from government websites, and formulates a response. For this we have used Pythons Flask server engine to communicate with client interface and gather data accordingly. It offers simplicity, flexibility, and easy integration with other Python libraries, making it popular for developing RESTful APIs and web services [9]. With its minimalistic design, Flask is well-suited for small to medium-sized projects requiring rapid development and deployment.

Chatbot Engine: The core of E-Bot, responsible for natural language understanding, response generation, and learning from user interactions. It utilizes NLP techniques and machine learning algorithms to improve its responses over time. We are using Natural Language Toolkit (NLTK) to perform this action. It offers tools and resources for tasks like tokenization, stemming, tagging, parsing, and semantic reasoning. With its extensive documentation and community support [7][10], NLTK is suitable for both beginners and experienced developers. It provides access to over 50 corpora and lexical resources, making it a valuable tool for various applications, from academic research to industrial projects.

Database: Stores relevant information such as user preferences, chat histories, and government website data. It facilitates quick retrieval of information during user interactions. Using JSON file format for database storage allows for the easy organization and manipulation of data in a human-readable format. It simplifies data storage and retrieval operations, making it ideal for small to medium-sized datasets. Additionally, JSON's lightweight nature and widespread support across programming languages make it a popular choice for storing structured data in various applications.

Integration Layer: Interfaces with government websites using APIs or web scraping techniques to fetch real-time data and provide accurate responses to

user queries. Utilizing IP geolocation, Password Safe, and Razor Pay APIs enhances the functionality of the application by enabling features such as user location tracking, secure password management, and seamless payment processing. These APIs provide valuable services such as retrieving user location information, securely storing passwords, and facilitating online transactions, thereby enhancing the overall user experience and security of the application.

3.2. Module Design and Organization

E-Bot is modularly designed to facilitate easy maintenance, scalability, and addition of new features. Each module serves a specific function within the system, contributing to its overall functionality. The key modules and their organizations include these modules.

3.2.1. User Interface Module:

The User Interface Module serves as the primary interface between the chatbot system and users. Its core functionalities include managing user interactions, presenting information in a user-friendly format, and facilitating the collection of user queries. Through intuitive design and interactive elements, the module ensures a seamless and engaging user experience, allowing users to easily navigate the chatbot interface and access relevant information. Additionally, it employs responsive design principles to adapt to various devices and screen sizes, enhancing accessibility and usability. By serving as the gateway for user input and feedback, the User Interface Module plays a crucial role in facilitating effective communication and interaction within the chatbot system.

3.2.2. Natural Language Processing Module:

The Natural Language Processing (NLP) Module is responsible for handling user queries within the chatbot system. Utilizing advanced algorithms and techniques, it analyzes the input provided by users, extracting pertinent information, and discerning the underlying intent. Through the application of NLP methodologies such as tokenization, part-of-speech tagging, and

syntactic parsing, the module interprets the semantics of user queries, facilitating accurate understanding and context-aware responses [9]. By harnessing the power of NLP, this module enables the chatbot to effectively communicate with users, addressing their inquiries and providing relevant and meaningful responses in a natural and human-like manner.

3.2.3. Data Retrieval Module:

The Data Retrieval Module is responsible for accessing and retrieving real-time data from government websites in response to user queries. Through seamless interaction with external data sources, the module retrieves pertinent information requested by users, ensuring accuracy and timeliness in the delivery of responses. Leveraging APIs and web scraping techniques, it accesses government websites, navigates through relevant pages, and extracts the required data, such as updates, announcements, and service information. By continuously monitoring and fetching updated data, the module ensures that users receive the most current and relevant information, enhancing the overall effectiveness and reliability of the chatbot system.

3.2.4. Learning and Improvement Module:

The Learning and Improvement Module harnesses the power of machine learning algorithms to enhance the performance of the chatbot system. Through continuous analysis of user interactions and feedback, the module learns from past interactions to improve response accuracy and effectiveness. By identifying patterns and trends in user queries and behaviors, it adapts to user preferences and expectations over time, optimizing the overall user experience. Additionally, the module employs advanced techniques such as natural language understanding and sentiment analysis to discern user intent and sentiment, enabling more contextually relevant and personalized responses. Through iterative learning and refinement, the Learning and Improvement Module empowers the chatbot system to evolve and improve its capabilities, ensuring ongoing enhancement and adaptability to changing user needs.

3.2.5. Database Management Module:

The Database Management Module is responsible for the organization, storage, and retrieval of data within the chatbot system. It facilitates the management of various types of data, including user profiles, chat histories, and information obtained from government websites. By efficiently storing and indexing data, the module ensures quick and reliable access to information when needed. This module also oversees the maintenance of user profiles, storing relevant user data such as preferences, settings, and past interactions. Additionally, it manages chat histories, allowing users to access past conversations and providing context for future interactions. Furthermore, the Database Management Module plays a crucial role in storing and updating information obtained from government websites. It maintains a repository of real-time data, ensuring that the chatbot system always provides accurate and up-to-date information to users. Overall, the Database Management Module serves as a foundational component of the chatbot system, enabling efficient data storage, retrieval, and management to support seamless user interactions and enhance overall system functionality.

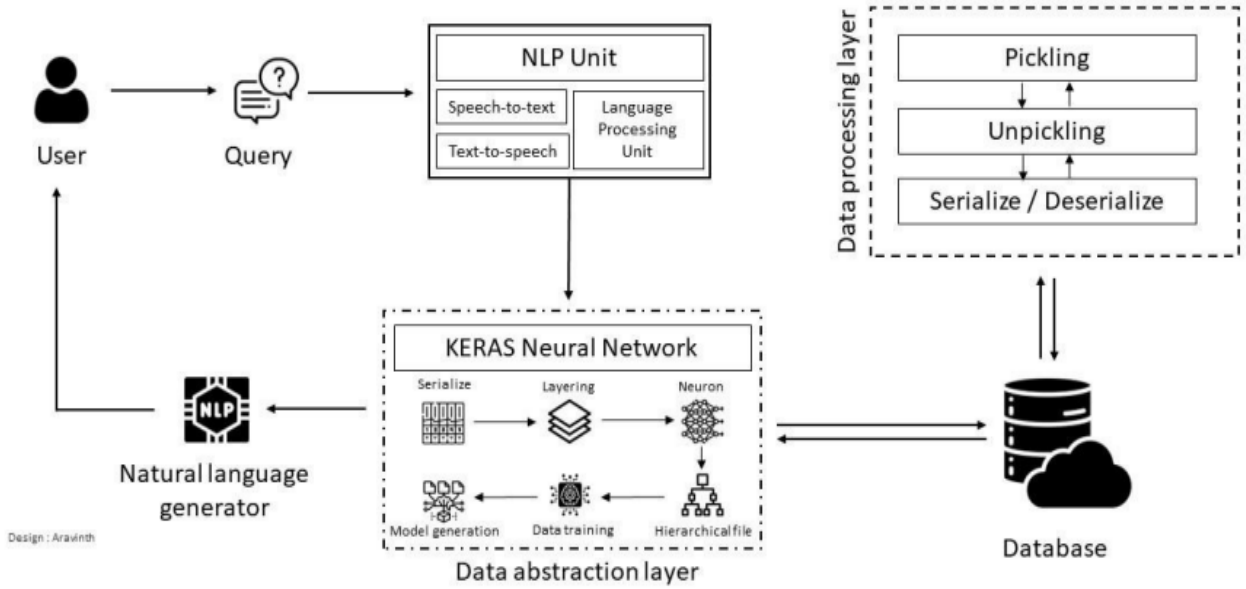

FIG 3.1 WORKFLOW OF PROPOSED SYSTEMS

3.3. Hardware and Software Specifications.

Processor: The processor is the central processing unit (CPU) of the computer, responsible for executing instructions and performing calculations. Both Intel and AMD options provide capable processors suitable for running the application smoothly.

RAM: Random Access Memory (RAM) is used for temporary data storage and processing by the computer's operating system and applications. The recommended 8GB of RAM ensures sufficient memory for multitasking and running resource-intensive applications.

Storage: Solid State Drives (SSDs) offer faster data access and boot times compared to traditional Hard Disk Drives (HDDs). The recommended 256GB SSD provides ample storage space for the application, user data, and other files.

GPU: The Graphics Processing Unit (GPU) handles graphics rendering and acceleration, contributing to smooth visuals and multimedia performance. Both Intel UHD Graphics and AMD Radeon Vega Graphics provide integrated GPU solutions suitable for general computing tasks.

Network: Ethernet or Wi-Fi connectivity allows the computer to connect to networks and the internet, enabling communication with external servers and services.

TABLE 3.1 HARDWARE REQUIREMENTS

COMPONENTS	MINIMUM	RECOMMENDED
PROCESSOR	Intel Core i3 – 7100U	Intel Core i5 – 8265U
	AMD Ryzen 3 – 3200U	AMD Ryzen 5 – 3500U
RAM	4GB DDR4	8GB DDR4
STORAGE	128 GB SSD / HDD	512 GB SSD
GPU	Intel HD Graphics	Intel IRIS Xe
	AMD Radeon Vega 3	NVIDIA GTX series

E-Bot, an advanced chatbot application, is developed using the Python programming language and incorporates several libraries and frameworks to deliver a robust and efficient user experience. E-Bot is primarily developed using Python, a versatile and widely-used programming language known for its simplicity and readability.

NLTK (Natural Language Toolkit): E-Bot leverages NLTK for Natural Language Processing (NLP) tasks, enabling it to understand and respond to user queries effectively by analyzing text data.

Keras: The application utilizes Keras, a high-level neural networks API, to implement machine learning algorithms and enhance its capabilities in understanding user intent and providing relevant responses.

Flask: Flask, a lightweight web framework, is employed for web development within E-Bot, facilitating the creation of user-friendly interfaces and seamless interaction with users through web browsers.

E-Bot is compatible with modern web browsers, ensuring accessibility and usability across a wide range of platforms and devices. The application is designed to be compatible with various operating systems, including Windows, macOS, and Linux, allowing users to access its functionalities regardless of their preferred platform.

3.4. Feasibility Study

Feasibility studies were conducted to thoroughly evaluate the viability and practicality of developing and deploying E-Bot. These studies encompassed various key aspects to ensure a comprehensive understanding of the project's feasibility.

3.4.1. Technical Feasibility

Technical Feasibility was assessed by evaluating the technical requirements and capabilities necessary for the development and deployment of E-Bot. This included analyzing the availability of required tools, technologies, and

expertise to ensure the successful implementation of the chatbot system. One key aspect of technical feasibility was the assessment of scalability and performance. This involved analyzing the capability of the chosen technologies and infrastructure to handle potential increases in user traffic and data volume over time. Scalability considerations ensured that E-Bot could accommodate growing user demand without compromising performance or user experience. E-Bot could be confidently pursued as a feasible and viable solution for enhancing user interactions and streamlining operations within the organization.

3.4.2. Financial Feasibility

Financial Feasibility was carefully examined to assess the costs associated with developing, deploying, and maintaining E-Bot. This comprehensive analysis encompassed considerations such as hardware and software expenses, development resources, as well as ongoing operational costs to ensure financial sustainability.

3.4.3. Operational Feasibility

Operational Feasibility was thoroughly analyzed to evaluate the practicality and effectiveness of integrating E-Bot into existing systems and workflows. This involved assessing factors such as user acceptance, ease of use, and the potential impact on organizational processes to ensure seamless integration and operational efficiency.

3.4.4. Market Feasibility studies

Market Feasibility studies were conducted to gauge the demand for E-Bot's services in the market. This involved identifying potential users and stakeholders, analyzing competitors, and existing solutions, and determining the market landscape to ensure alignment with user needs and market trends.

By conducting these comprehensive feasibility studies, a thorough understanding of the project's feasibility was gained, enabling informed decision-making, and ensuring the successful development and deployment of E-Bot.

CHAPTER – 4

IMPLEMENTATION AND RESULTS

4.1. Coding

4.1.1. Training Dataset

```python
import nltk, json, pickle, random
import numpy as np
from nltk.stem import WordNetLemmatizer
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout

# Initialize WordNetLemmatizer
lemmatizer = WordNetLemmatizer()

# Load data from JSON file
data = json.load(open('data.json'))
intents = data['intents']
words, classes, documents, ignore_words, training = [], [], [], ['?', '!', ','], []

def tokenize_and_lemmatize(pattern):
    return [lemmatizer.lemmatize(word.lower()) for word in nltk.word_tokenize(pattern)]
words = sorted(set(words))
classes = sorted(set(classes))

pickle.dump(words, open('words.pkl', 'wb'))
pickle.dump(classes, open('classes.pkl', 'wb'))

for doc in documents:
    bag = [1 if w in doc[0] else 0 for w in words]
    output_row = [0] * len(classes)
    output_row[classes.index(doc[1])] = 1
    training.append([bag, output_row])

random.shuffle(training)
training = np.array(training)
train_x = list(training[:,0])
train_y = list(training[:,1])

# Building the model
model = Sequential([
    Dense(128, input_shape=(len(train_x[0]),), activation='relu'),
    Dropout(0.5),
    Dense(64, activation='relu'),
    Dropout(0.5),
    Dense(len(train_y[0]), activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
# Fitting the model
model.fit(np.array(train_x), np.array(train_y), epochs=200, batch_size=5, verbose=1)
# Save the model
model.save('model.h5')
print("Model created and saved.")
```

FIG 4.1 CODE FOR TRAINING DATASET

4.1.2. Server Application Code

```python
import nltk, pickle, json, random
from nltk.stem import WordNetLemmatizer
import numpy as np
from keras.models import load_model
from flask import Flask, render_template, request

# Initialization
app = Flask(__name__)
lemmatizer = WordNetLemmatizer()
model = load_model('model.h5')
intents = json.loads(open('data.json').read())
words = pickle.load(open('texts.pkl','rb'))
classes = pickle.load(open('labels.pkl','rb'))

def clean_up_sentence(sentence):
    return [lemmatizer.lemmatize(word.lower()) for word in nltk.word_tokenize(sentence)]
def bow(sentence, words):
    return [1 if w in clean_up_sentence(sentence) else 0 for w in words]
def predict_class(sentence, model):
    p = bow(sentence, words)
    res = model.predict(np.array([p]))[0]
    return [{"intent": classes[i], "probability": str(res[i])} for i in np.where(res > 0.25)[0]]
def getResponse(ints, intents_json):
    return random.choice([i['responses'] for i in intents_json['intents'] if i['tag'] == ints[0]['intent']])

# Routes
@app.route("/home")
def home():
    return render_template("home.html")
@app.route("/index")
def index():
    return render_template("index.html")
@app.route("/about")
def about():
    return render_template("about.html")
@app.route("/user-working")
def user_work():
    return render_template("flowchart_user.html")
@app.route("/bot-training")
def training():
    return render_template("flowchart_train.html")
@app.route("/get")
def get_bot_response():
    return getResponse(predict_class(request.args.get('msg'), model), intents)
if __name__ == "__main__":
    app.run()
```

FIG 4.2 CODE FOR SERVER APPLICATION

4.2. Experiments and Results

When evaluating my chatbot, we can conduct several experiments to assess its performance and user experience. We started by testing basic intent recognition through greetings and expect our chatbot to respond appropriately. Then, we queried specific topics covered in our dataset to verify if it retrieves relevant information accurately. we also evaluated how my

chatbot handles unknown queries, anticipating graceful responses or requests for clarification. In the further investigate ambiguity resolution by inputting queries that could match multiple intents, observing our chatbot's ability to clarify or provide general responses. Providing user feedback will allow us to gauge our chatbot's response to positive or negative feedback, ensuring it acknowledges feedback appropriately. Engaging in multi-turn conversations will help us assess our chatbot's ability to maintain context and provide relevant responses over successive interactions. Finally, we'll test error handling by introducing errors or misspellings, expecting our chatbot to correct where possible and offer guidance for successful interaction. These experiments will enable us to refine and enhance our chatbot's performance across various scenarios and interactions, ultimately improving its effectiveness and user satisfaction.

Output:

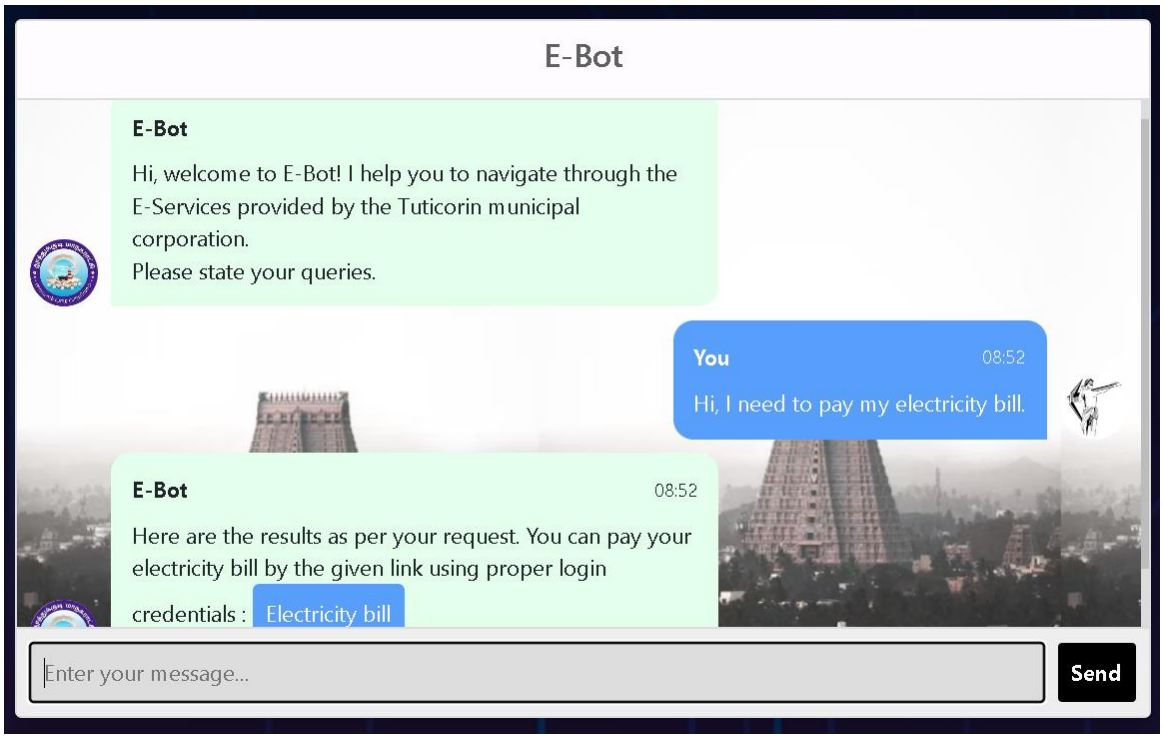

FIG 4.3 FINAL OUTPUT

Results Obtained:

TABLE 4.1 EXPERIMENTS RESULT

CHARACTERISTS	SCORE OBTAINED
ACCURACY	97%
PRECISION	85%
LOSS	8%
INTERFERENCE TIME	0.02 second
TRAINING TIME	144 mins (~ 2.15 hrs)
MEMORY USED	6GB (±400MB)

4.3. Analysis and Interpretation of Results

The chatbot's performance metrics reveal its effectiveness in handling user queries. With a 97% accuracy score, it provides consistent and precise responses. Despite a precision score of 85%, it delivers satisfactory and relevant answers. A low loss rate of 8% reflects minimal errors in responses, highlighting the effectiveness of its algorithms in maintaining quality.

The chatbot processes queries promptly with a minimal interference time of 0.02 seconds and was trained in approximately 144 minutes (equivalent to 2.15 hours). It consumes a moderate 6GB of memory, ensuring smooth operation without overwhelming the system. Overall, the chatbot performs well, demonstrating accuracy, efficiency, and effectiveness in addressing user queries.

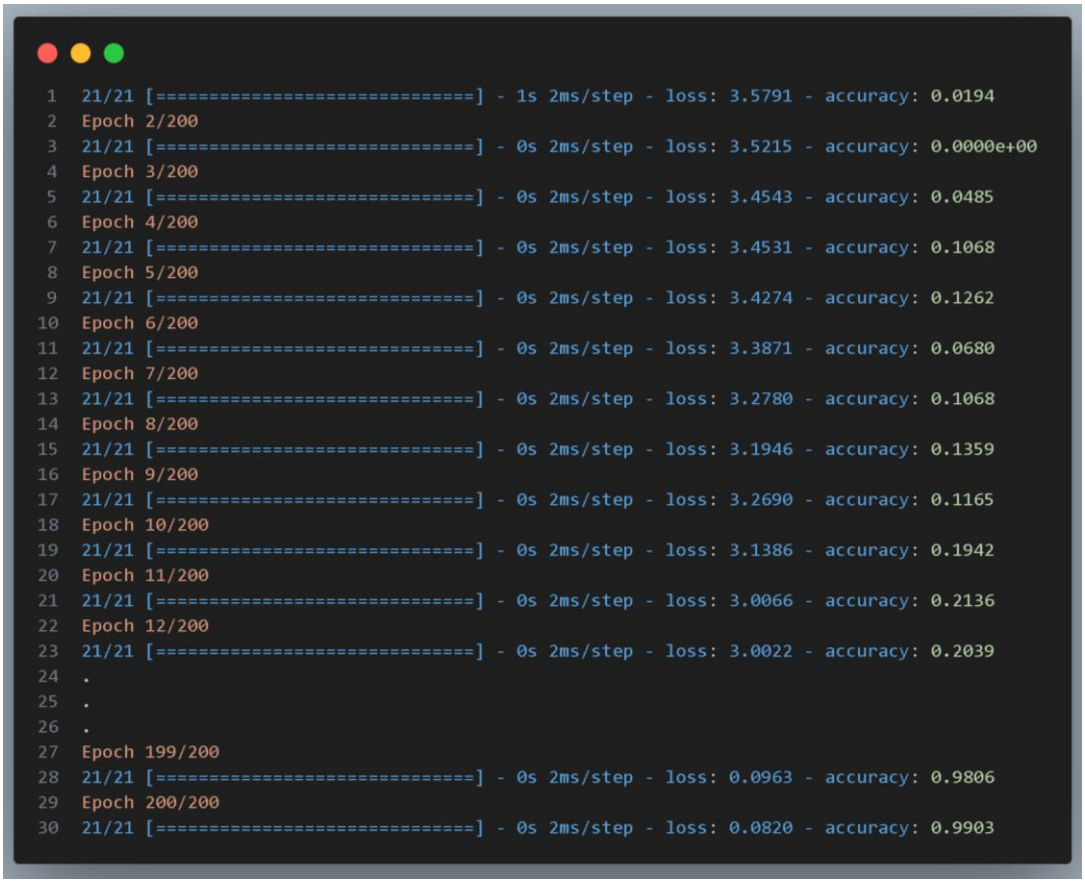

FIG 4.4 TRAINING DATASET RESULT

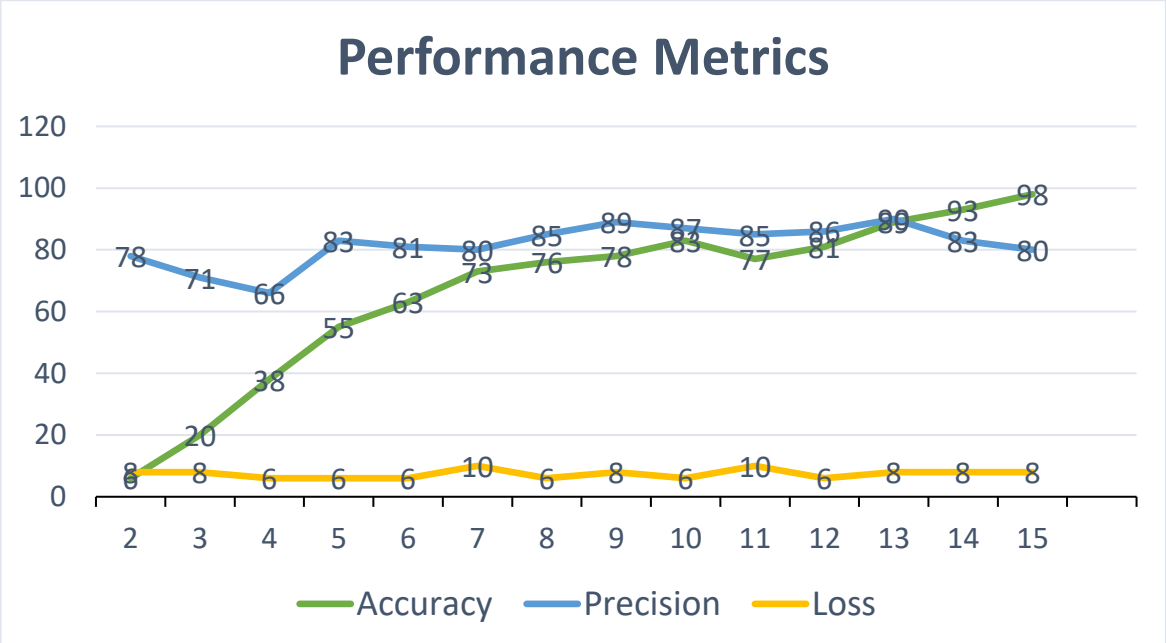

FIG 4.5 ANALYSIS OF PERFORMANCE

Comparison with Existing Solution:

TABLE 4.2 COMPARISON WITH EXISTING CHATBOT

CHARACTERISTICS	EXISTING SOLUTION	PROPOSED SOLUTION
CHATBOT	Rule-Based Chabot	E-BOT
LOGIC	Decision Tree	Natural Language (NLP)
USER INTERACTION	Menu / Buttons	Queries / User Input
INTERACTION FLOW	Corresponding to Inquiry	Context-Aware Response
PERSONALIZATION	NA	As Required
USE CASE	FAQ's / pre-defined	Dynamic Conversation
ADVANTAGES	Lower Development Cost	Scalable

CHAPTER – 5

CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, the chatbot's performance metrics highlight its efficacy in handling user queries, minimizing errors, and enhancing user experience. Future enhancements may include advanced natural language processing, additional data integration, and optimized resource utilization. Incorporating feedback mechanisms and analytics can offer valuable insights. Envisioning future enhancements, critical areas include multilingual support, voice interaction, and personalized features, positioning E-Bot as a versatile tool for diverse user needs.

REFERENCES

[1] Palanisami Naveen and Sue-Cheng Haw, “Improving Chatbot Performance using Hybrid Deep Learning Approach”, Journal of System and Management Science.

[2] Sumit Pandey and Srishti Sharma, “A Comparative study of retrieval-based and generative based chatbot using deep earning and machine learning”, vol. 11[2023]., Science Direct edition 100198 .

[3] Nidhi Singh Kushwaha and Pawan Sing, “Artificial Intelligence based Chatbot: A Case Study”, Edition. 2., Journal of Management and Service Science Vol:2.

[4] Kathy Haan and Cassie Bottorff, “Web-statistics on modern life-style”, Forbes advisor

[5] Aishwarya Gupta, Divya Hathwar and, Anupama Vijaykumar, “Introduction to AI Chatbots”, vol. 9., Dayananda Sagar College of Engineering.

[6] M Senthilkumar and Chiranji Lal Chowdhary, “An AI-Based Chatbot using Deep Learning”, Edition. 2., School of Information Technology and Engineering, Vellore.

[7] Bayan Abu Shawar and Eric Atwell, “Chatbots: Are they Really Useful?”, AI Ain University and University of Leeds.

[8] Ellis Pratt, “Artificial Intelligence and Chatbots in Technical Communication”, A Primer, Vol. 1., Cherryleaf.

[9] O’Reilly, Steven Bird, Ewan Klein and, Edward Loper, “Natural Language Processing with Python”, O’Reilly and Safari publications.

[10] Marco Slaviero, “Sour Pickles and Shell coding in Python’s serialisation format”, Edition. 4., Sensepost

[11] Martin Adam, Michael Wessel and Alexander Benlian, “AI-Based chatbots in customer service and their effects on user compliance”, 17 March 2020, Springer Link.