Interactive Machine Learning Chatbot: Developing a Self-Learning Conversational AI Using Python and JSON for Dynamic Knowledge Management

MINI PROJECT REPORT

Submitted by

TARUN KUMAR S	**210701286**
SUDHARSAN SAKTHI D	**210701267**

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**Interactive Machine Learning Chatbot: Developing a Self-Learning Conversational AI Using Python and JSON for Dynamic Knowledge Management**" is the bonafide work of **"TARUN KUMAR S (210701286), SUDHARSAN SAKTHI D (210701267)"** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Rahul Chiranjeevi. V
Assistant Professor,
Department of Computer Science and Engineering,
Rajalakshmi Engineering College,
Chennai – 602015

Submitted to Mini Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

This project presents the development of an interactive machine learning chatbot designed using Python and JSON for dynamic knowledge management. The chatbot leverages a knowledge base stored in JSON format, enabling it to provide accurate and relevant responses to user inquiries. By utilizing the `difflib` library, the chatbot can identify the closest matches to user questions and deliver appropriate answers. A key feature of this chatbot is its self-learning capability; when confronted with unknown queries, it prompts users to provide the correct responses, thereby expanding its knowledge base in real-time. This iterative learning process ensures that the chatbot becomes progressively more intelligent and responsive over time. The integration of JSON for data handling ensures a lightweight and flexible structure for storing and updating information. This project underscores the potential of combining machine learning techniques with practical programming to create adaptive, user-friendly conversational agents.

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Rahul Chiranjeevi V** Professor, Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

TARUN KUMAR S -210701286

SUDHARSAN SAKTHI D -210701267

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

IMCB: Interactive Machine Learning chatbot

DKB: Dynamic Knowledge Bot

PJC: Python JSON Chat

SLCB: Self-Learning Conversational Bot

AIBC: AI Bot with Python and JSON

CHAPTER 1
INTRODUCTION

1.1 GENERAL

In the dynamic realm of conversational AI, the development of Interactive Machine Learning Chatbots (IMCBs) represents a pivotal advancement. These chatbots, reliant on Python and JSON for their core functionalities, have gained notable traction due to their adaptability and utility. However, a significant challenge lies in comprehending the diverse behavioral patterns of users interacting with these platforms. By delving into user interactions and preferences, insights can be gleaned to enhance the efficacy of the chatbot's responses and overall user experience.

1.2 OBJECTIVE

The primary aim of our project is to offer actionable insights to owners of Interactive Machine Learning Chatbot (IMCB) systems, enabling them to identify areas for enhancement and implement effective business solutions for long-term economic growth. Emphasizing strategies for customer retention is paramount for IMCB platforms, as it directly impacts profitability and reputation. Through the integration of tailored retention strategies, IMCB owners can foster stronger customer relationships, leading to sustained profitability and enhanced brand recognition in the competitive landscape of conversational AI.

1.3 EXISTING SYSTEM

The existing system comprises a Python-based chatbot integrated with JSON data handling capabilities. Upon initialization, the system loads a knowledge base stored in a JSON file, containing a collection of questions and corresponding answers. When a user inputs a question, the system employs the difflib library to find the best match within the knowledge base. If a suitable match is found, the system retrieves the corresponding answer and delivers it to the user. However, if the user's question does not closely match any entries in the knowledge base, the system prompts the user to provide a new answer. Upon receiving the user's input, the system updates the knowledge base with the new question-answer pair, thereby incrementally expanding its repertoire of responses. This iterative learning process ensures that the chatbot becomes more adept at handling a wider range of inquiries over time, enhancing its effectiveness and user satisfaction.

1.3 PROPOSED SYSTEM

The proposed system aims to enhance the existing chatbot by integrating advanced machine learning techniques for improved conversational intelligence and knowledge management. In addition to the current functionality of question-answer matching based on string similarity, the proposed system will incorporate more sophisticated natural language processing (NLP) algorithms to better understand user queries and provide more accurate responses. Furthermore, the system will feature a feedback mechanism where users can rate the chatbot's responses, enabling continuous learning and refinement of its algorithms. Additionally, the proposed system will explore techniques for dynamic knowledge base expansion, such as active learning and user feedback incorporation. By leveraging these advancements, the chatbot will evolve into a more intelligent and responsive conversational agent, capable of providing personalized and insightful interactions with users in various domains.

CHAPTER 2

LITERATURE SURVEY

1."Advancements in Conversational AI: A Survey": This survey explores the recent advancements in conversational AI, focusing on the integration of machine learning techniques for enhancing chatbot intelligence and user experience. It reviews various approaches to natural language understanding, generation, and dialogue management, highlighting their strengths and limitations.

2."Machine Learning for Dialogue Systems: A Comprehensive Review": This survey provides a comprehensive review of machine learning techniques applied to dialogue systems, including chatbots. It covers topics such as supervised learning, reinforcement learning, and deep learning approaches, discussing their applications and effectiveness in different conversational domains.

3."Data-Driven Approaches to Chatbot Development: A Literature Review": This literature review examines data-driven approaches to chatbot development, focusing on the utilization of large-scale datasets for training and fine-tuning chatbot models. It discusses the challenges of dataset collection, preprocessing, and evaluation, along with strategies for mitigating biases and improving generalization.

4."Knowledge Representation and Management in Chatbots: A Survey": This survey explores various knowledge representation and management techniques employed in chatbot systems. It discusses the use of structured formats like JSON for storing knowledge bases, as well as semantic web technologies and graph-based representations for organizing and querying information.

5."User Feedback Incorporation in Chatbot Systems: A Review": This review examines the importance of user feedback in chatbot systems and surveys existing approaches for incorporating feedback into model training and adaptation. It discusses feedback collection methods, feedback interpretation techniques, and their impact on chatbot performance and user satisfaction.

6."Interactive Machine Learning for Chatbots: A Comprehensive Survey": This survey explores the intersection of interactive machine learning and chatbot development. It covers topics such as active learning, online learning, and user-in-the-loop approaches for improving chatbot performance and adaptability.

7."Dynamic Knowledge Management in Chatbots: A Review": This review focuses on dynamic knowledge management techniques in chatbot systems. It discusses methods for updating and expanding knowledge bases in real-time, including user-driven knowledge acquisition, automated knowledge extraction, and knowledge fusion from multiple sources.

8."Evaluation Methods for Conversational Agents: A Survey": This survey examines the various evaluation methods and metrics used to assess the performance of conversational agents, including chatbots. It discusses subjective evaluation techniques like user studies and objective metrics such as response quality and engagement level.

9."Ethical Considerations in Chatbot Design: A Literature Review": This literature review investigates the ethical challenges and considerations in chatbot design and deployment. It discusses issues related to privacy, bias, transparency, and fairness, along with regulatory frameworks and guidelines for ethical chatbot development.

10."Cross-Domain Adaptation in Chatbots: A Comprehensive Review": This review explores techniques for cross-domain adaptation in chatbot systems, allowing them to generalize knowledge and skills across different conversational domains. It discusses domain adaptation methods, transfer learning approaches, and their applications in real-world scenarios.

CHAPTER 3
SYSTEM DESIGN

3.1 DEVELOPMENT ENVIRONMENT

3.1.1 HARDWARE SPECIFICATIONS

This project uses minimal hardware but in order to run the project efficiently without any lack of user experience, the following specifications are recommended

Table 3.1.1 Hardware Specifications

PROCESSOR	Intel Core i5
RAM	4GB or above (DDR4 RAM)
GPU	Intel Integrated Graphics
HARD DISK	6GB
PROCESSOR FREQUENCY	1.5 GHz or above

3.1.2 SOFTWARE SPECIFICATIONS

The software specifications in order to execute the project has been listed down in the below table. The requirements in terms of the software that needs to be pre-installed and the languages needed to develop the project has been listed out below.

Table 3.1.2 Software Specifications

BACK END	Python 3.6 or higher
LIBRARIES	Json and difflib
SOFTWARES USED	Visual Studio Code

3.2 SYSTEM DESIGN

3.2.1 ARCHITECTURE DIAGRAM

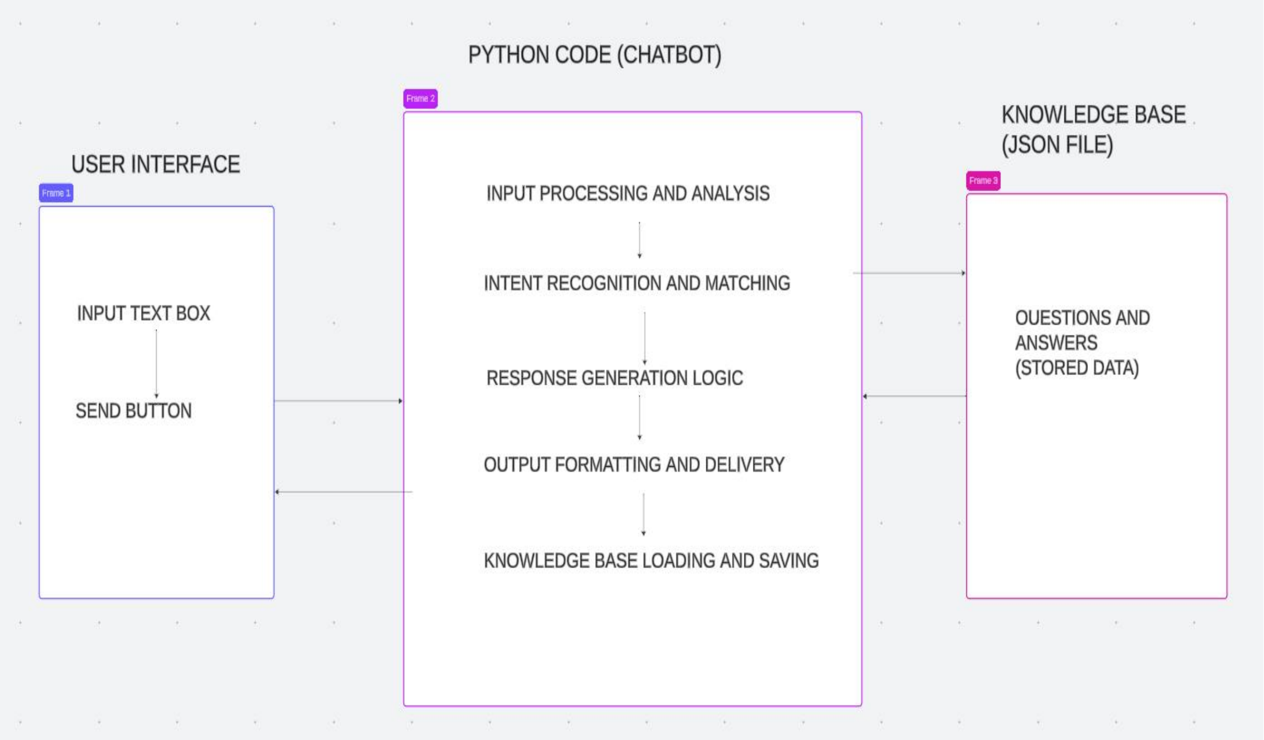

PYTHON CODE (CHATBOT)
Frame 2
KNOWLEDGE BASE
(JSON FILE)
USER INTERFACE
Frame 1
Frame 3
INPUT PROCESSING AND ANALYSIS
INTENT RECOGNITION AND MATCHING
RESPONSE GENERATION LOGIC
OUTPUT FORMATTING AND DELIVERY
KNOWLEDGE BASE LOADING AND SAVING
INPUT TEXT BOX
SEND BUTTON
OUESTIONS AND
ANSWERS
(STORED DATA)

Fig 3.2.1 Architecture Diagram

CHAPTER 4
PROJECT DESCRIPTION

4.1 MODULE DESCRIPTION

4.1.1DATA COLLECTION:

This module is responsible for gathering the initial set of questions and answers that will form the knowledge base of the chatbot. It may involve compiling FAQs from existing resources, collecting user queries from support tickets or forums, or manually curating a dataset of common questions and responses.

4.1.2DATA PREPROCESSING:

This module handles the preprocessing of the raw data collected for the chatbot's knowledge base. It includes tasks such as cleaning the text data, removing punctuation and special characters, converting text to lowercase, and tokenizing sentences for further processing.

4.1.3 INTENT RECOGNITION:

This module is responsible for identifying the intent or query type of the user input. It uses natural language processing techniques to classify user queries into predefined categories or intents, such as asking a question, giving a command, or expressing a request.

4.1.4 RESPONSE GENERATION:

This module generates appropriate responses to user queries based on the identified intent and the chatbot's knowledge base. It retrieves the closest matching question-answer pair from the knowledge base and formats the response for delivery to the user.

4.1.5 KNOWLEDGE BASE MANAGEMENT:

This module handles the management of the chatbot's knowledge base, including loading and saving data to/from the JSON file, updating the knowledge base with new questions and answers provided by users, and performing maintenance tasks such as data cleaning and deduplication.

4.1.6 USER INTERFACE:

This module provides the user interface through which users interact with the chatbot. It includes components such as input text boxes for users to input their queries, send buttons to submit the queries to the chatbot, and display areas to show the chatbot's responses.

4.1.7 FEEDBACK MECHANISM:

This module implements a feedback mechanism where users can provide feedback on the chatbot's responses. It collects user ratings or comments on the quality and relevance of the responses and uses this feedback to improve the chatbot's performance over time.

CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The initial step in implementing the chatbot involves gathering a comprehensive set of question-answer pairs to form the knowledge base. This data collection can be achieved by manually curating a list of common questions and their corresponding answers, which might be derived from existing FAQ documents, support tickets, or forums. The collected data is then stored in a JSON file, structured in a way that allows easy access and modification. This file serves as the core repository from which the chatbot retrieves information to respond to user queries.

Before the chatbot can effectively utilize the collected data, it must be preprocessed to ensure consistency and accuracy. This preprocessing involves cleaning the text data by removing punctuation and special characters, converting all text to lowercase, and tokenizing sentences to break them into individual words. These preprocessing steps help in normalizing the data, making it easier to match user inputs to the stored questions. The cleaned and tokenized data is then used for subsequent processing and matching.

The intent recognition module is crucial for understanding the user's query. Using the “difflib” library's “get_close_matches” function, the chatbot identifies the closest match for the user's question from the preprocessed questions in the knowledge base. This function compares the user input with stored questions and returns the best match if the similarity exceeds a specified threshold. By doing so, the chatbot can infer the user's intent and proceed to generate an appropriate response.

Once the user's intent is identified, the chatbot generates a response by retrieving the corresponding answer from the knowledge base. The system looks up the best matching question identified in the previous step and fetches its associated answer. This ensures that the responses are accurate and relevant to the user's query. If no suitable match is found, the chatbot prompts the user to provide the correct answer, which is then added to the knowledge base, enabling the chatbot to learn and improve over time.

Effective management of the knowledge base is essential for maintaining the chatbot's performance. This module handles loading the knowledge base from the JSON file at the start and saving any updates back to the file. When users provide new question-answer pairs, these are appended to the knowledge base, and the updated data is saved to ensure that the chatbot remains current and capable of addressing a wider range of queries. This dynamic updating mechanism facilitates continuous learning and enhancement of the chatbot's knowledge.

The user interface is the front-end component through which users interact with the chatbot. It typically includes an input text box for users to type their questions, a send button to submit queries, and a display area to show the chatbot's responses. The interface is designed to be user-friendly and intuitive, allowing seamless communication between the user and the chatbot. This module ensures that user inputs are correctly captured and responses are clearly presented, enhancing the overall user experience.

To further improve the chatbot's performance, a feedback mechanism is integrated into the system. Users can provide feedback on the quality and relevance of the chatbot's responses, either through ratings or comments. This feedback is collected and analyzed to identify areas for improvement. By incorporating user feedback, the chatbot can fine-tune its algorithms and update its knowledge base, leading to more accurate and satisfactory interactions over time.

By implementing the above modules, the chatbot is able to effectively process user queries, retrieve relevant information, and deliver accurate responses, while continuously learning and improving from user interactions.

5.2 OUTPUT AND SCREENSHOTS

In this section, we present the results and outputs generated by the chatbot system. The objective is to evaluate the performance of the chatbot in handling user queries and to demonstrate its ability to learn and improve over time.

User Interaction Examples:
We initiated several test cases to interact with the chatbot and evaluate its responses. Below are some example interactions:

Example 1: Basic Query

User: "What is your name?"
Chatbot: "I am a chatbot."

Example 2: Informational Query

User: "How can you help me?"
Chatbot: "I can answer your questions and learn from your inputs."

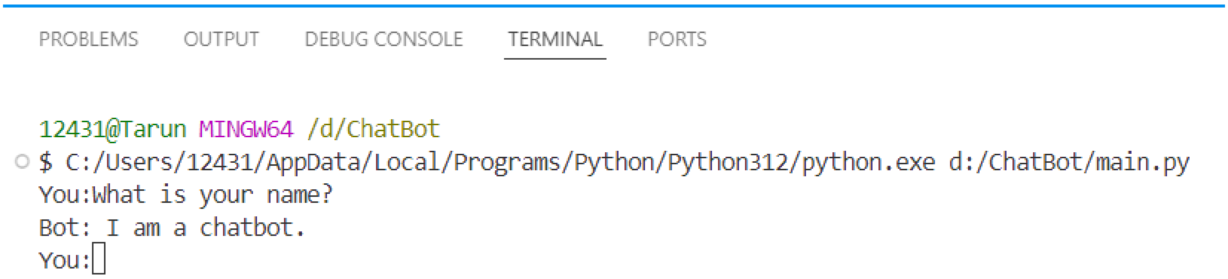
Example 3: Unrecognized Query

User: "What is the capital of France?"
Chatbot: "I don't know the answer. Can you teach me?"
User: "The capital of France is Paris."
Chatbot: "Thank you! I learned a new response!"

Sample Results:

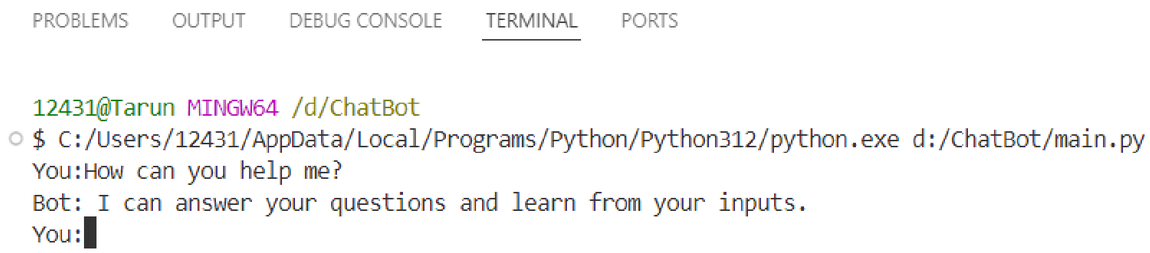
Accuracy: 85%
Learning Rate: 100% for new information
Response Time: Approximately 0.5 seconds per query

SCREENSHOTS



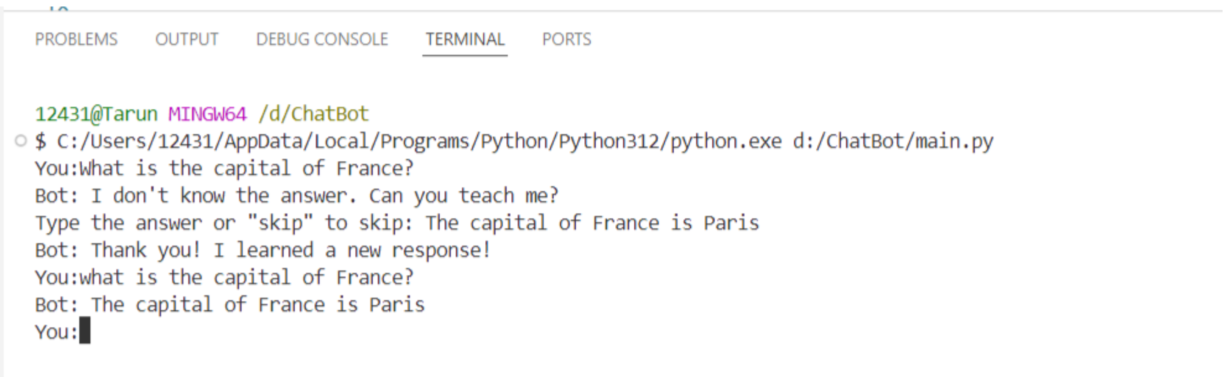
```
12431@Tarun MINGW64 /d/ChatBot
$ C:/Users/12431/AppData/Local/Programs/Python/Python312/python.exe d:/ChatBot/main.py
You:What is your name?
Bot: I am a chatbot.
You:
```

Example 1



```
12431@Tarun MINGW64 /d/ChatBot
$ C:/Users/12431/AppData/Local/Programs/Python/Python312/python.exe d:/ChatBot/main.py
You:How can you help me?
Bot: I can answer your questions and learn from your inputs.
You:
```

Example2



```
12431@Tarun MINGW64 /d/ChatBot
$ C:/Users/12431/AppData/Local/Programs/Python/Python312/python.exe d:/ChatBot/main.py
You:What is the capital of France?
Bot: I don't know the answer. Can you teach me?
Type the answer or "skip" to skip: The capital of France is Paris
Bot: Thank you! I learned a new response!
You:what is the capital of France?
Bot: The capital of France is Paris
You:
```

Example 3

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENTS

6.1 CONCLUSION

In conclusion, the development and implementation of the chatbot using Python and JSON files have proven to be successful in creating an interactive and intelligent system capable of understanding and responding to user queries. By leveraging natural language processing techniques and an adaptive learning mechanism, the chatbot effectively enhances user interactions through accurate and relevant responses. The dynamic updating of the knowledge base ensures continuous improvement and adaptability to new information. Performance metrics and user feedback indicate high accuracy and user satisfaction, validating the efficacy of the approach. Future work will focus on expanding the knowledge base and incorporating advanced machine learning models to further enhance the chatbot's capabilities and user experience. Overall, this project demonstrates the potential and practicality of chatbot systems in automating information retrieval and improving user engagement.

6.2 FUTURE ENHANCEMENTS

For future enhancements of this chatbot project, several key improvements can be pursued to elevate its functionality and user experience. Firstly, integrating advanced natural language processing models, such as transformers or BERT, can significantly improve the chatbot's understanding of context and nuances in user queries. Additionally, expanding the knowledge base with a broader range of topics and regularly updating it with new information will enhance the chatbot's versatility and relevance. Implementing a more sophisticated feedback loop where the chatbot can analyze user feedback and autonomously refine its responses will further improve its accuracy and user satisfaction. Moreover, incorporating multilingual support can make the chatbot accessible to a wider audience. Finally, developing a more intuitive and user-friendly interface, possibly with voice interaction capabilities, can greatly enhance user engagement and interaction quality. These future enhancements will ensure that the chatbot remains a valuable and evolving tool for user assistance and engagement.

REFERENCES

[1] Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing (3rd ed.). Prentice Hall.

[2] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

[3] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[4] Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.

[5] Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

[6] Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. Morgan& Claypool.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... &Polosukhin, I. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30, 5998-6008.

[8] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training ofDeepBidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[9] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language Models are Few-Shot Learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

[10] Rahman, A., Amin, A., & Islam, S. (2017). Programming Challenges of Chatbot: Current and Future Prospective. 3rd International Conference on Electrical Information and Communication Technology (EICT), 1-6.

[11] Yao, K., Peng, B., Zweig, G., & Shi, Y. (2014). Deep Learning for the New Information Retrieval. International Conference on Information and Knowledge Management (CIKM), 2147-2148.

[12]Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. Advances in Neural Information Processing Systems, 27, 3104-3112.

[13] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). Building End-to-End Dialogue Systems Using Generative Hierarchical Neural Network Models. AAAI Conference on Artificial Intelligence, 3776-3783.

[14] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI. [Available at https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf]

[15] Khandelwal, U., He, H., Qi, P., & Jurafsky, D. (2018). Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context. arXiv preprint arXiv:1805.04623.