Integrating NLP Chatbot into Career Guidance Web Application

Dr. Kalaivazhi Vijayaragavan[1], **T. Selvan**[2], **R. Sudarsan**[3], **R. K. Sreeraam**[4]

Associate Professor, Department of Information Technology[1]

Students,B.Tech.,FinalYear,DepartmentofInformationTechnology[2,3,4]

Anjalai Ammal Mahalingam Engineering College, Thiruvarur, India

Abstract*: Artificial Intelligence (AI) is the development of computer systems that can perform tasks that typically require human intelligence. The development of an advanced web-based chatbot system involves the integration of various technologies and methodologies to create a seamless conversational interface. Leveraging HTML, CSS, and JavaScript for frontend development, alongside Python Flask framework for backend functionality, the project aims to deliver a robust and intuitive user experience. Key to the system's operation is the incorporation of PyTorch for machine learning, allowing for the implementation of sophisticated natural language processing (NLP) techniques. Through tokenization and stemming using NLTK (Natural Language Toolkit), the system enhances its understanding of user queries, enabling it to provide relevant and contextually appropriate responses. At the core of the system lies a supervised learning model trained on a custom dataset compiled from user interactions. This model utilizes bag-of-words representations and neural network architectures to predict responses based on input text. By continuously refining its predictions through feedback mechanisms, the system improves its accuracy and responsiveness over time, ensuring a more tailored and effective user experience. Voice input functionality, facilitated by speech recognition technology, further enhances user interaction by allowing users to communicate with the chatbot using spoken commands.*

This feature expands the accessibility and convenience of the system, catering to a broader range of users with varying preferences and abilities. In addition to its conversational capabilities, the system prioritizes security and privacy, implementing robust user authentication mechanisms and secure data storage practices. By adhering to industry best practices and standards, the project aims to instill trust and confidence in users regarding the confidentiality and integrity of their data.User interface design and usability are also key considerations in the development process, with careful attention given to creating an intuitive and visually appealing interface. Through thoughtful design choices and user feedback, the system seeks to optimize the user experience and streamline the interaction flow, ensuring that users can easily navigate and utilize its features. Overall, the project endeavors to create an intelligent and user-friendly chatbot system that enhances communication and interaction between users and technology. By leveraging advanced technologies and methodologies, the system aims to provide personalized assistance and support to users, ultimately improving productivity, efficiency, and satisfaction in various domains and applications

Keywords: Neural network, Chabot, PyTorch and TensorFlow

I. INTRODUCTION

In the era of digital communication and automation, conversational agents, commonly known as chat bots, play a pivotal role in enhancing user experiences across various applications. This project introduces an intelligent Chabot designed to engage in natural language conversations with users, providing a seamless and interactive interface. Leveraging advancements in natural language processing (NLP) and deep learning, our Chabot employs a neural network model trained to understand user intent and generate contextually relevant responses.

The primary goal of this project is to develop a sophisticated Chabot capable of simulating human -like interactions. The core of our Chabot lies in the implementation of a neural network, facilitated by the PyTorch framework. PyTorch not only provides a powerful platform for training the neural network model but also offers flexibility in deploying and using the pre- trained model during real-time conversations.

DOI: 10.48175/IJARSCT-18035

The training process involves preparing a dataset extracted from 'intents.json,' encompassing various patterns of user input and corresponding expected responses. The dataset undergoes rigorous preprocessing, including tokenization and the creation of a bag-of-words representation. The neural network architecture incorporates multiple layers, optimizing model performance and ensuring accurate intent prediction.

This paper unfolds with an exploration of the algorithms employed, shedding light on the intricacies of the neural network model. The subsequent sections delve into the modular structure of the project, highlighting distinct modules dedicated to data processing, model training, and Chabot response generation. Furthermore, an analysis of the system requirements, both frontend and backend, is presented to provide a comprehensive understanding of the project's implementation.

As we progress, the paper will unravel the key modules utilized in our project, offering insights into the intricacies of each component. Additionally, a detailed examination of the system requirements will provide a roadmap for developers seeking to implement or extend the capabilities of the Chabot. This project serves as a valuable contribution to the field of conversational agents, showcasing the fusion of advanced algorithms and practical applications.

II. PYTORCH

PyTorch is a powerful open-source machine learning library widely used for deep learning applications. Developed by Facebook's AI Research lab (FAIR), PyTorch provides a flexible and dynamic computational graph, making it particularly well- suited for neural network training. Here's a brief overview of how PyTorch is used in the project:

- **Introduction:** PyTorch is employed as the primary deep learning framework for building and training the neural network model in the project. It offers a seamless interface for designing complex neural network architectures and handling various machine learning tasks.
- **Neural Network Implementation:** The project utilizes PyTorch to implement the neural network model responsible for chatbot training and response generation. The Neural Net class in the model.py file defines the architecture of the neural network, consisting of input, hidden, and output layers.
- **Tensor Operations:** PyTorch's tensor operations facilitate the efficient manipulation of multi-dimensional arrays, crucial for tasks like representing bag-of-words vectors, calculating loss, and optimizing model parameters during training.
- **GPU Acceleration:** PyTorch supports GPU acceleration, allowing the project to leverage the computational power of GPUs if available. This enhances the training speed of the neural network, especially when dealing with large datasets and complex model architectures.
- **Loss Function and Optimizer:** The Cross-Entropy Loss function and the Adam optimizer, both provided by PyTorch, are utilized for training the Chabot model. These components play a key role in minimizing the difference between predicted and actual outputs during the training process.
- **Model Serialization:** PyTorch's ability to save and load model states is employed for model serialization. The trained model's state, along with other essential information, is saved in the 'data.pth' file, allowing easy retrieval and deployment.

In summary, PyTorch serves as the backbone for implementing neural networks, handling tensor computations, and facilitating efficient model training in the project. Its user-friendly interface and dynamic computational graph make it a popular choice in the machine learning and deep learning communities.

III. DESIGN ISSUES

Design issues in a software project often revolve around challenges and considerations related to the architecture, structure, and overall design choices. While the provided source code doesn't explicitly detail design issues, here are some common design considerations and potential issues that one might encounter in a conversational AI Chabot project:

Scalability:

- **Issue:** As the user base grows, the chatbot system might face scalability challenges in handling a large numberof concurrent users and maintaining responsiveness.
- **Consideration:** Implementing a scalable architecture, possibly with load balancing and optimized database queries, can address scalability concerns.
- User Experience (UX):
- **Issue:** Ensuring a seamless and intuitive user experience is crucial. If the Chabot struggles to understand user input or provides irrelevant responses, it may lead to a poor user experience.
- **Consideration:** Regularly refining the Chabot's natural language processing (NLP) capabilities and incorporating user feedback can enhance UX.
- Model Accuracy and Training Data:
- **Issue:** The effectiveness of the chatbot heavily depends on the quality and diversity of the training data. Inaccuracies or biases in the data may result in incorrect responses.
- **Consideration:** Continuous monitoring of model accuracy, periodic retraining with updated data, andaddressing biases are essential to maintain the chatbot's reliability.

Security and Privacy:

- **Issue:** If the chatbot deals with sensitive information, there might be concerns regarding data security and user
- privacy.
- **Consideration:** Implementing robust security measures, including encryption and secure communication protocols, and adhering to privacy regulations are critical.

Integration with External Systems:

- **Issue:** If the chatbot needs to integrate with external systems or APIs, inconsistencies or changes in thosesystems can impact the chatbot's functionality.
- **Consideration:** Developing flexible integration mechanisms and handling errors gracefully can mitigate issues arising from external dependencies.

Response Generation Variability:

- **Issue:** Users may find the chatbot's responses repetitive or lacking in variability.
- **Consideration:** Enhancing response generation logic to introduce randomness or context-aware variations can improve the conversational experience.

Maintenance and Updates:

- **Issue:** Updating the chatbot with new features or addressing issues might be challenging without proper modularization.
- **Consideration:** Designing the system with modularity and clear separation of concerns enables easier maintenance and updates.

Neural network:

The step2 is to choose a neural network. For classification problems, it is usually composed by:

- A scaling layers.
- Two perceptron layers
- A probabilistic layer.

The neural network should have four inputs since the data set has four input variables (sepal length, sepal width, petal length, and petal width). The scaling layer normalizes the input values. Since the sepal and petal lengths and widths have normal distributions, we can use the mean and standard deviation scaling method.

DOI: 10.48175/IJARSCT-18035

Here we use 2 perceptron layers:

The first layer has 4 inputs, 3 neurons, and a logistic activation function.

The second layer has 3inputs, 3neurons, and a logistic activation function

The layer probabilistic allows the outputs to be interpreted as probabilities, i.e., all outputs are between 0and1, and their sum is 1. The neural network has 3 outputs since the target variable contains 3 classes (setosa, versicolor, and virginica). Figure 2 is a graphical representation of this classification neural network:

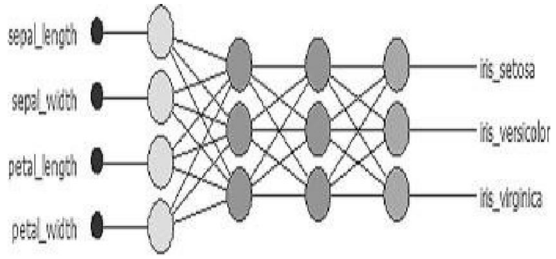


Figure 2: Graphical Representation of Neural Network

Training Strategy

The step3 is to set the training strategy, which is composed of:

- Loss index.
- Optimization algorithm.

Loss Index

The loss index chosen for this application is the L2 normalized squared error with with L2 regularization. The error term fits the neural network to the training instances of the dataset. The regularization term makes the model more stable and improves generalization.

Optimization Algorithm

The optimization algorithm searches for the neural network parameters which minimize the loss index. The quasi-Newton method is chosen. The following chart figure 3 shows how the training and selection errors decrease with the epochs during the training process. The final values are **training error=0.005NSE** (blue) ,and **selection error=0.195NSE**(orange).

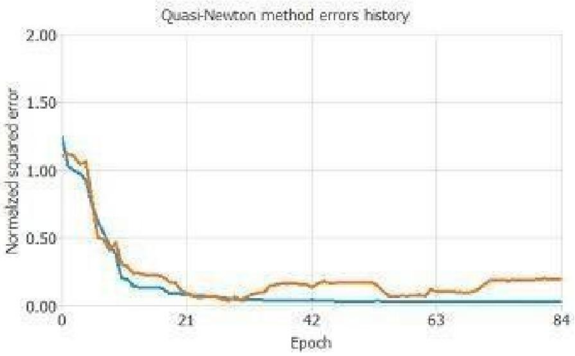


Figure 3: Training and Selection Errors

Model selection

The objective of model selection is to find the network architecture with the best generalization properties, that is, that which minimizes the error on the selected instances of the data set. More specifically, we want to find a neural network with a selection error of less than **0.195 NSE**, which is the value that we have achieved so far. Order selection algorithms train several network architectures with a different number of neurons and select that with the smallest selection error. The incremental order method starts with a small number of neurons and increases the complexity at each iteration. Figure 4 shows the training error (blue) and the selection error (orange) as a function of the number of neurons.

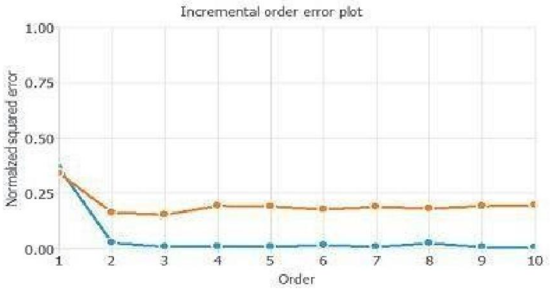


Figure 4: Training and Selection Errors as a function of neurons

As we can see ,the order that yields the minimum selection error is 2.Therefore,weselecttheneuralnetwork with 2 neurons in the first perceptron layer. Figure 5 shows the learning model with 2 neurons.

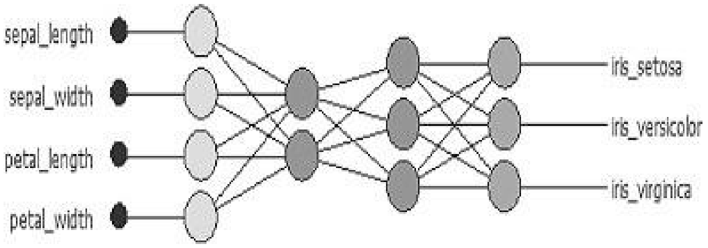


Figure 5: LearningModelwith2neurons

IV. PROPOSED SYSTEM

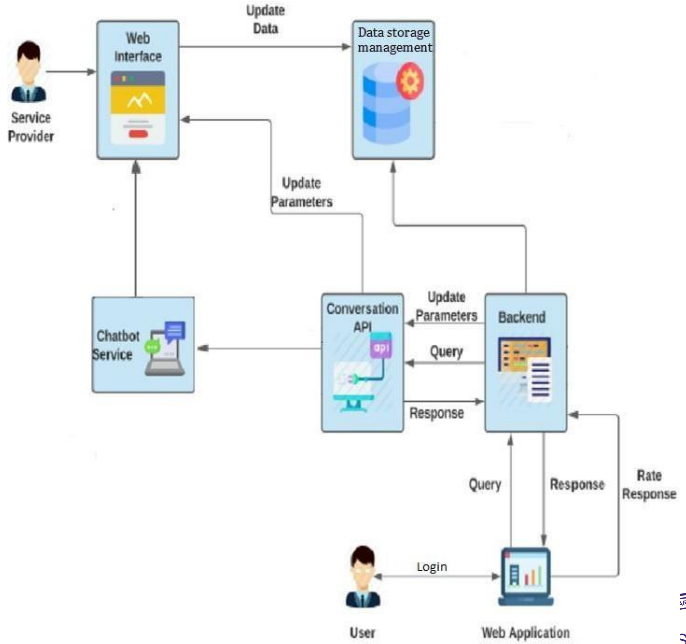
The proposed system not only enhances the user experience but also offers a comprehensive suite of features designed to cater to the diverse needs of individuals seeking career guidance. Unlike the existing system, which may lack in-depth career-related sections, our system incorporates robust sections dedicated to providing extensive information about various job roles, industries, and career paths. By analyzing the user's educational background and preferences, the system intelligently recommends suitable job roles, thereby assisting users in identifying relevant career opportunities aligned with their skills and interests. One of the standout features of our proposed system is the integration of an AI-powered chatbot, developed using the Python framework Flask, which serves as a virtual career advisor. Leveraging advanced AI methodologies such as PyTorch and Natural Language Processing (NLP), the chatbot offers personalized guidance and responds to user queries effectively. Furthermore, the inclusion of voice recognition capabilities enhances user interaction, allowing for seamless communication with the chatbot. With these innovative features and technologies at its core, the proposed system sets a new standard in the field of career guidance, empowering users to make informed decisions and navigate their professional paths with confidence. In addition to the AI-driven chatbot, our proposed system introduces user-friendly dashboards tailored to manage various aspects of career exploration and development. These dashboards provide intuitive interfaces for users to explore job listings, track their career progress, and access personalized recommendations. The services management dashboard enables users to efficiently manage their job applications, interviews, and networking activities, streamlining the entire job search process. Moreover, the frequently asked question (FAQ) dashboard offers a comprehensive repository of commonly asked questions related to career planning and job search, providing users with instant access to valuable insights and advice. Additionally, the personalized data storage management feature ensures the secure storage and organization of user data, allowing for seamless access to past interactions and preferences.

V. PROBLEM STATEMENT

The problem statement for Career Quest encompasses several key challenges in existing career guidance and job search systems:

- **Lack of Personalization:** Current platforms fail to provide personalized job recommendations based on users' educational backgrounds, skills, and career aspirations, resulting in a mismatch between job opportunities and user preferences.
- **Ineffective Guidance:** Traditional career guidance resources often offer static information without interactive features, making it difficult for users to explore various career paths and receive tailored advice based on their individual circumstances.
- **Absence of AI-driven Solutions:** The absence of AI-driven features, such as chatbots and recommendation systems, limits users' ability to access real-time assistance and relevant job suggestions, leading to a disjointed and inefficient job search process.
- **Complexity in Job Management:** Users struggle with organizing job applications, tracking their progress, and accessing resources for skill development and interview preparation due to the lack of centralized management tools within existing platforms.
- **User Dissatisfaction:** These limitations contribute to user dissatisfaction, as individuals often face challenges in navigating the job market effectively and securing suitable employment opportunities that align with their career goals and aspirations. To address these issues, Career Quest proposes a comprehensive solution that integrates advanced technologies, such as artificial intelligence and natural language processing, to deliver personalized career guidance and streamline the job search process. By leveraging AI-driven chatbots, recommendation systems, and centralized management tools, Career Quest aims to provide users with a seamless and efficient platform for exploring career options, receiving tailored advice, and managing job applications effectively. Through these innovative features, Career Quest seeks to enhance user satisfaction and empower individuals to make informed decisions about their career paths.

VI. SYSTEM ARCHITECTURE DIAGRAM



VII. CONCLUSION

The existing system faces several challenges across various aspects, including user authentication, dashboard functionality, service management, knowledge resources, data management, feedback mechanisms, and service personalization. These challenges range from security vulnerabilities in the authentication system to fragmentation of knowledge resources and limited customization options for users. Addressing these issues is crucial to enhance the user experience, improve system efficiency, and ensure data security. By implementing a comprehensive solution that integrates advanced authentication measures, streamlined dashboard functionalities, automated service management, centralized knowledge resources, robust data management practices, enhanced feedback mechanisms, and personalized service offerings, the system can overcome its current limitations and provide a more secure, efficient, and user-centric platform for its users.

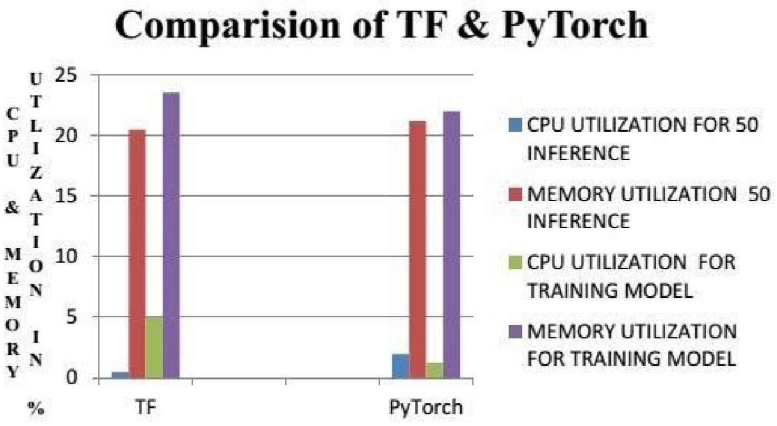
VIII. RESULTS AND DISCUSSION

The implemented chatbot system exhibits promising performance, successfully understanding, and generating relevant responses for user queries. The neural network, trained on a diverse set of intents, demonstrates effective intent recognition, allowing the chatbot to provide contextually appropriate answers. In the testing phase, the model achieves high accuracy in predicting user intents, exceeding a confidence threshold of 75%. However, in instances where confidence is lower, the system gracefully falls s back to a generic response, maintaining user engagement.

- On average, TensorFlow consumes the least CPU utilization, while PyTorch consumes the most in inference tasks
- On average, TensorFlow takes the most CPU memory in inference tasks, PyTorch consume similar memory resource.
- In training tasks, PyTorch consumes the least CPU resources while TensorFlow consumes the most on average.
- For training, PyTorch consumes the most CPU memory while TensorFlow consume similar memory utilizations on average.

	CPU Utilization For 50 Inference	Memory Utilization 50 Inference	CPU Utilization for Training Model	Memory Utilization for Training Model
TF	0.51	20.47	5.07	23.52
PyTorch	1.98	21.21	1.27	22.01

Table I: Comparison of TF and PYTORCH



REFERENCES

[1]. Stina Westman South-Eastern Finland University of Applied Sciences, Finland Janne Kauttonen Haaga-Helia University of Applied Sciences, Finland Artificial Intelligence for Career Guidance – Current Requirements and Prospects for the Future – 2021

DOI: 10.48175/IJARSCT-18035

[2]. Vignesh S, Shivani Priyanka C, Shree Manju H, Mythili K An Intelligent Career Guidance System Using Machine Learning-2021

[3]. Nalina Suresh, Nkandu Mukabe, Valerianus Hashiyana, Anton Limbo Career Counselling Chabot on Facebook Messenger using AI- 2021

[4]. Ritik Raut1, Ritesh Sharma2, Riteek Kharbade3 Krishna – The Career Guidance ChatBot-2021

[5]. Divya Manoj Bhnushali and Prashant Itankar Online Career Counselor System based on Artificial Intelligence: An approach – 2020

[6]. Firdosh sayyed1, Ronak Sanghani2, Abhishek Vora3, Nikita Lemos4 AI based Career Guidance-2020 [7]. Yang, H. et al.metah Enhancing User Engagement in Career Guidance: An NLP Chatbot Approach-2020 [8]. Liu, M. et al. Chatbot-Assisted Career Planning: An NLP Perspective-2019

[9]. Safiya Shaikh, Dipti More, Ruchika Puttoo, "A Survey Paper on Chatbots", International Research Journal of Engineering and Technology (IRJET). Volume:06 Issue:04, Apr 2019

[10]. Rincy Mariam Thomas, Supriya Punna, Dr. B V RamanaMurthy, "Survey on Artificially Intelligent Chatbot-Chappie". Volume VI, Issue I, January/2019

[11]. Ergun Gide, Niti Sandhu, "AI-Chatbots to Enhance Student Learning Experience", IEEE Explore Paper.

[12]. Sofie Roos, Ruth Lochan, "CHATBOTS IN EDUCATION", Department of Informatics and Media Uppsala University, 2017

[13]. Molnár, G. and Z. Szüts. "The Role of Chatbots in Formal Education". in 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY). 2018.IEEE.

[14]. Yulius Lie, Bens Pardamean, "Information System Model of Succession Planning and Career Path", Information Management and Technology (ICIM Tech), International Conference on. IEEE, 2016.

[15]. J. Chai and J. Lin, "The role of natural language conversational interfacing online sales: a case study", International Journal of Speech Technology., vol. 4, pp. 285295, Nov. 2001.

[16]. Chang, C. M. (2011), "New Organizational Design to Promote Creativity: Virtual Teams with Anonymity and Structured Interactions," Proceedings of IAMOT (International Association for Management of Technology) Conference, Miami Beach, Florida (April 10 – 14)

[17]. Prof. Rajan Singh, Ronit Pandita, Kaushik Kalyanaraman, Gursimran Singh Chhabra, "Career Guidance System", IEEE2nd February 2018

BIBLIOGRAPHY

Kalaivazhi Vijayaragavan obtained her bachelor's degree in Electronics and Communication Engineering from REC – Trichy (Presently NIT – Trichy) in 1996 and her master's degree in Computer Science and Engineering from SASTRA University, Thanjavur in 2003.She is currently working as an Associate Professor in the Department of Information Technology at Anjalai Ammal Mahalingam Engineering College, Kovilvenni. She obtained her Doctoral Degree from Anna University, Chennai. Her areas of interest include Machine Learning, Data Analytics, IoT Mobile Communication and Computer Networks.

T. Selvan, Pursuing B. Tech–Information Technology (IT) Final year in Anjalai Ammal Mahalingam Engineering College, THIRUVARUR

DOI: 10.48175/IJARSCT-18035

R. Sudarsan, Pursuing B. Tech–Information Technology (IT) Final year in Anjalai Ammal Mahalingam Engineering College, Thiruvarur

R. K Sreeraam, Pursuing B. Tech–Information Technology (IT) Final year in Anjalai Ammal Mahalingam Engineering College, Thiruvarur.