ONGC Summer Training Project Report

Topic: Build a ChatBot

Guided By

Mr. Seemanta Das

Deputy General Manager (Programming) at
ONGC, Dehradun

Submitted By

Md Ishtiyaque Ahsan

B. Tech Computer Engineering

Zakir Husain College of Engineering and
Technology, AMU, Aligarh, UP

Training project

Introduction

Introduction to the Project: ONGC ChatBot

The project undertaken at ONGC represents the development of a chatbot system, exclusively designed to enhance the interaction between ONGC and its stakeholders, including employees, customers, and partners. This training project aims to provide comprehensive knowledge and hands-on experience with the new chatbot, designed to streamline communication, provide instant support, and improve operational efficiency within ONGC.

The primary objective of this training project is to equip participants with the skills and knowledge necessary to effectively use and manage the ONGC Chatbot. By the end of this training, participants will be able to interact with the chatbot, understand its various features and functionalities, and leverage its capabilities to ensure a seamless user experience.

Key Features of the ChatBot:

1. User-Friendly Interface: The chatbot is designed with a simple and intuitive interface to ensure ease of use for all users, accessible through web and mobile platforms.
2. Real-Time support: Provides instant responses to queries, offering 24/7 support for a wide range of topics, including ONGC services, employee queries, and customer support.
3. Efficient Query Management**:** Capable of handling, managing, and resolving user queries efficiently, with options to escalate complex issues to human agents when necessary.
4. Comprehensive Reporting and Analytics: Generates detailed reports on user interactions, common queries, and usage statistics to help ONGC understand user needs and improve services.
5. Security and Privacy: Ensures robust security measures and compliance with data protection regulations to safeguard user information and maintain privacy.

Technology used

The development of the ONGC chatbot involved a combination of machine learning, natural language processing, and web technologies to create an efficient and responsive system. Below is a detailed list of the technologies and tools used:

1. Programming Languages and Frameworks

- **Python:** The primary programming language used for developing the chatbot. Python's simplicity and extensive libraries make it ideal for AI and ML applications.
- **PyTorch:** An open-source machine learning library used for developing and training the neural network model.
- **Tkinter:** A standard GUI library in Python used to create the user interface for the chatbot application.

2. Natural Language Processing (NLP) Tools

- **nltk:** The Natural Language Toolkit is a library used for processing human language data. It includes functionalities for tokenizing, stemming, and bag-of-words processing.
- **JSON:** Used to store and manage the intents and responses for the chatbot in a structured format.

3. Machine Learning and Deep Learning

- **Neural Networks:** A feedforward neural network was used, implemented using PyTorch. The model consists of input, hidden, and output layers with ReLU activation functions.
- **CrossEntropyLoss:** Used as the loss function for training the neural network, which is suitable for classification problems.
- **Adam Optimizer:** An optimization algorithm used to update the weights of the neural network during training.

4. Data Handling and Processing

- **NumPy:** A library for numerical computations in Python, used for handling arrays and performing mathematical operations.
- **Dataset and DataLoader (PyTorch):** Utilized to manage and load training data efficiently during the training process.

5. Model Training and Evaluation

- **Training Script (train.py):** Includes the entire pipeline for loading data, processing it, training the model, and saving the trained model state.
- **Model Definition (model.py):** Defines the architecture of the neural network used in the chatbot.

6. Deployment and Integration

- **Chat Script (chat.py):** Handles the interaction between the user and the chatbot. It loads the trained model, processes user inputs, and generates responses.
- **torch.save and torch.load:** Functions used to save and load the trained model state for deployment.

7. Graphical User Interface (GUI)

- **Tkinter Application (app.py):** Provides a user-friendly interface for the chatbot, enabling users to interact with the chatbot through a simple GUI.

9. Intents Management

- **intents.json:** A JSON file used to define the various intents, patterns, and responses the chatbot can handle. This file is crucial for training the model and handling user queries.

8. Utilities

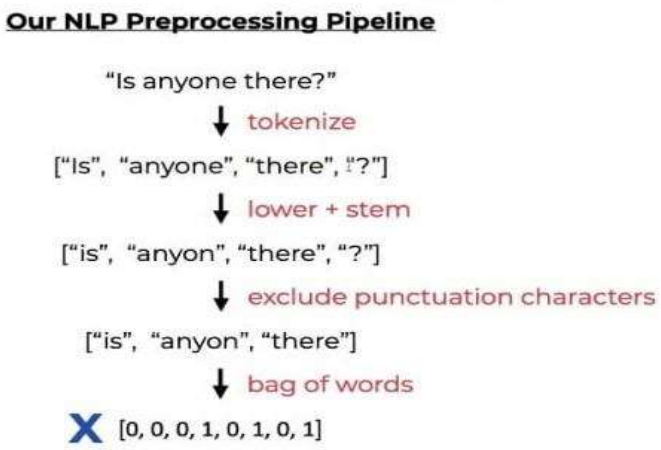
nltk_utils.py: Custom utility functions for NLP tasks such as tokenization, stemming, and creating

Approach and Implementation:

- ## Bag of Words from a given sentence:

The bag of words (BoW) model is a fundamental technique in natural language processing used to transform textual data into numerical form, making it suitable for machine learning algorithms. Below is a detailed approach to creating a bag of words model and its implementation in the ONGC chatbot project.

Following procedure is taken to get bag of words from a given sentence:



1. Tokenization: splitting a string into meaningful units

   (e.g. words, punctuation characters, numbers)

"What would you do with this much money?"

- ["What", "would", "you", "do", "with", "this", "much", "money","?"]

2. Stemming: After tokenization, every word would be converted in lower case and stemming is done.

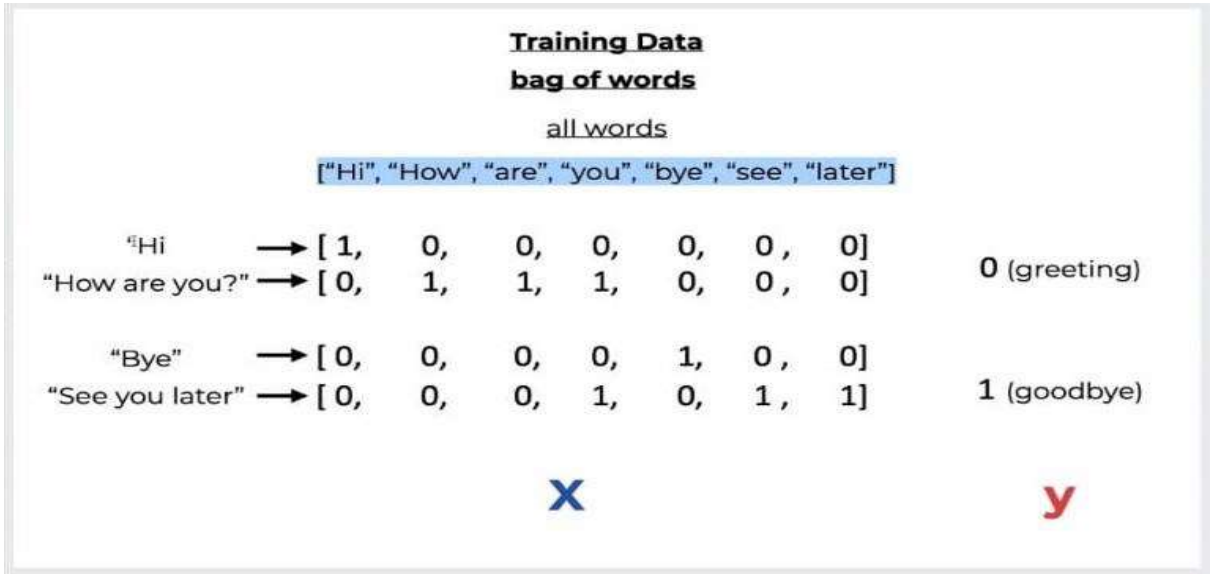
   In stemming, root form of the words is generated.

   e.g. – "organize", "organizing", "organizes"

   - "organ", "organ", "organ"

3. Bag of Words : first we get all_words from all the patterns,

- Then , for any sentence, BoW can be formed :



Here , in this way, any given sentence can be converted into a bag of words through the array of all words.

This BoW will then be used for implementing the model of chatbot.

- ## Format of Data which train the Model:

Intents data is in json format, which is used to train the chatbot model.
An element in intents.json is in 3 parts: tag, patterns, responses.

Tag: it uniquely identifies that intent. Intents having same tag will be considered as similar intents.

Patterns : It contains the questions that user can ask wrt a particular information.

Responses: It contains the responses that user will get fron bot for those questions.

Here is sample intent: {

```
      "tag": "EPINET_login",
      "patterns": ["How to Login to EPINET Portal?",
                   "Process followed to login to EPINET Portal."
      ],
      "responses": [
        "In order to access the E&P database login to EPINET portal (recommended
browser is Google chrome).\nEPINET Portal address EPINET
Site\nhttps://epinetddn.ongc.co.in Corporate EPINET Portal,
Dehradun\nhttps://epinetjrt.ongc.co.in EPINET, Jorhat\nhttps://epinetmum.ongc.co.in
EPINET, Mumbai\nhttps://epinetbrd.ongc.co.in EPINET,
Baroda\nhttps://epinetchn.ongc.co.in EPINET, Chennai\nhttps://epinetkol.ongc.co.in
EPINET, Kolkata\nLogin to the portal based on the approved user authentication.\nIf
user account not available, in order to create new login account, download the user
account creation from the EPINET\nHome page and submit the form duly approved to
the regional epinet center or mail the same to epinet mailing address."
      ]
    }
```

Here, tag is used to link questions(patterns) and answers(responses).

- ## Retrieval of intents.json from manual.pdf:

In the process of creating the intents.json file from the manual.pdf, we utilized Python scripts and natural language processing (NLP) techniques to convert descriptive text data into structured question-answer pairs suitable for chatbot training. Here is a detailed explanation of how this was achieved:

Tools and Libraries Used:

- **pdfplumber**: For extracting text from PDF files.
- **spaCy**: For performing NLP tasks such as sentence segmentation and named entity recognition.
- **re**: For regular expression operations to identify patterns in text.
- **json**: For saving the parsed data in JSON format.

Steps Involved:

1. **Extract Text from PDF**: We used `pdfplumber` to extract the text content from the PDF manual.
2. **Convert Text Data to Question-Answer Pairs**: The extracted text was then processed to convert descriptive sections into question-answer pairs. We used spaCy to segment sentences and identify key entities for generating questions.
3. **Save the Data in JSON Format**: The parsed question-answer pairs were saved into a JSON file, which could be used as the intents file for training the chatbot.

- **Implementation:**

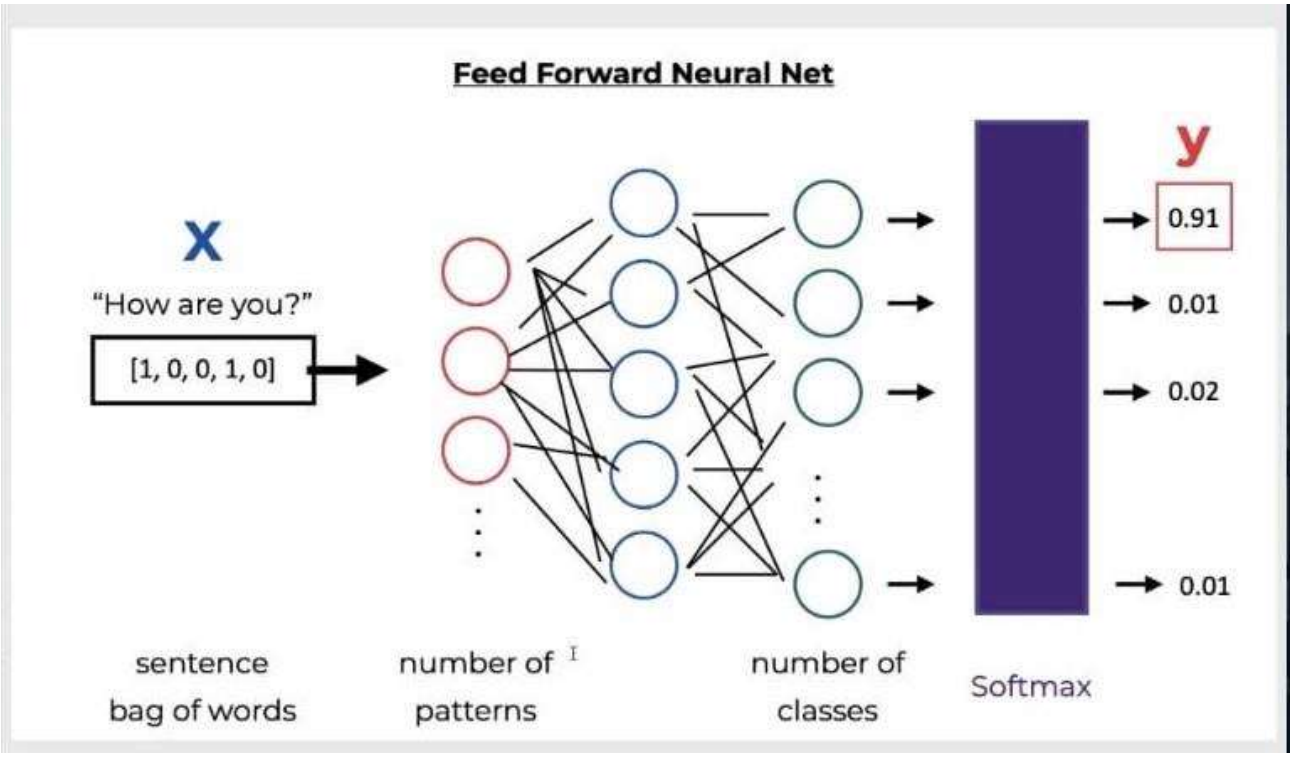
1. The Model analyses the intents.json, then collects all the words from patterns (questions) and stores them in an array (all_words). Then all_words is used to get bag of words from a sentence.

2. The Model also gets bag of words of all the patterns and then stores those BoWs with their respective tags in a pair i.e. pair of BoW and tag.
   All these pairs are stored in an array (lets say xy).

3. When user asks a query , then the query is converted into its BoW and then that BoW is compared with the BoWs stored in xy array. The BoW which matches more than 75% with user's query's BoW is the required intent and its tag is noticed.

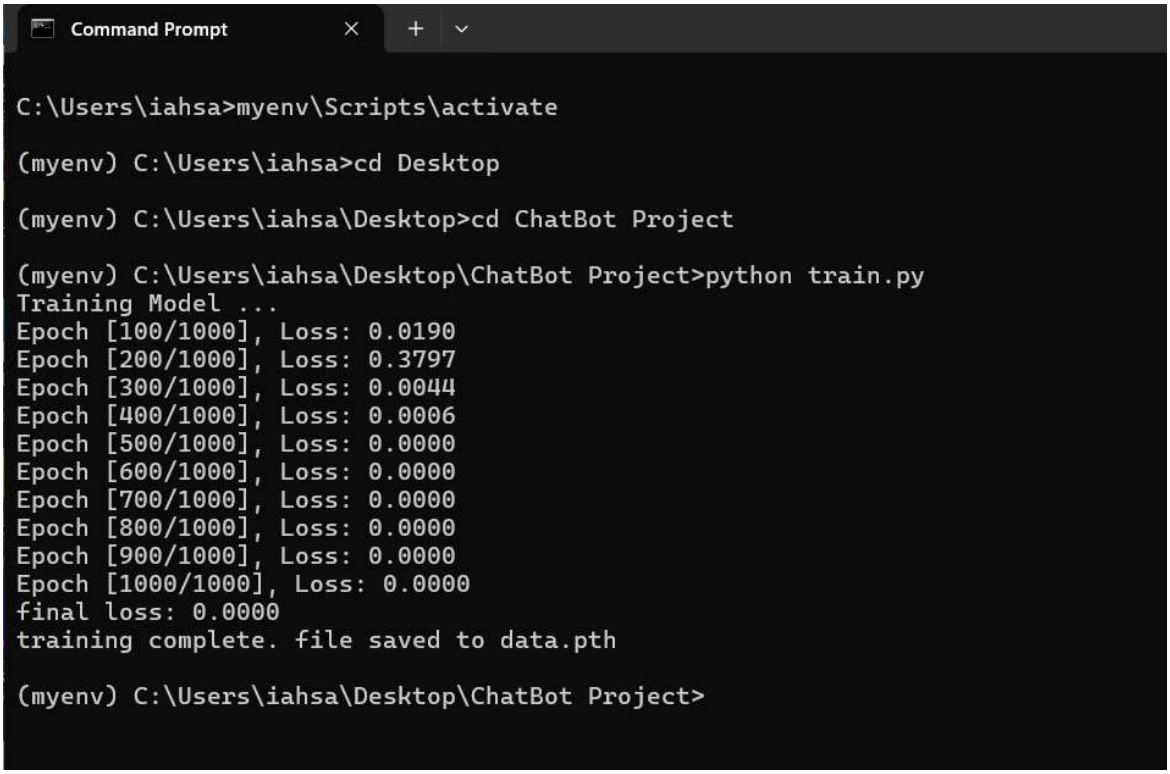
4. That intent's tag is used to identify that particular intent and the response related to that tag is sent as response to the query asked by user.

- Here we have the implementation of Feed Forward Neural net with 2 hidden layers.



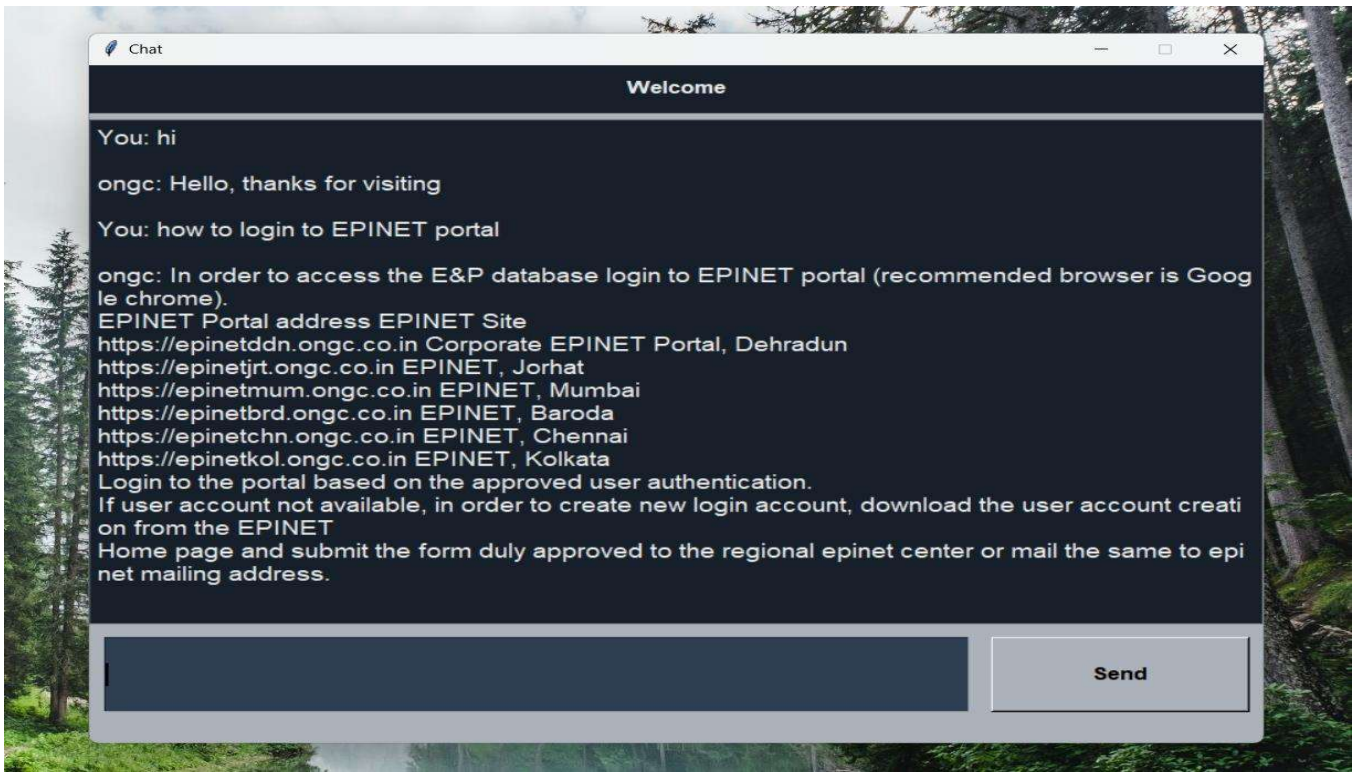
- ## Project Result:

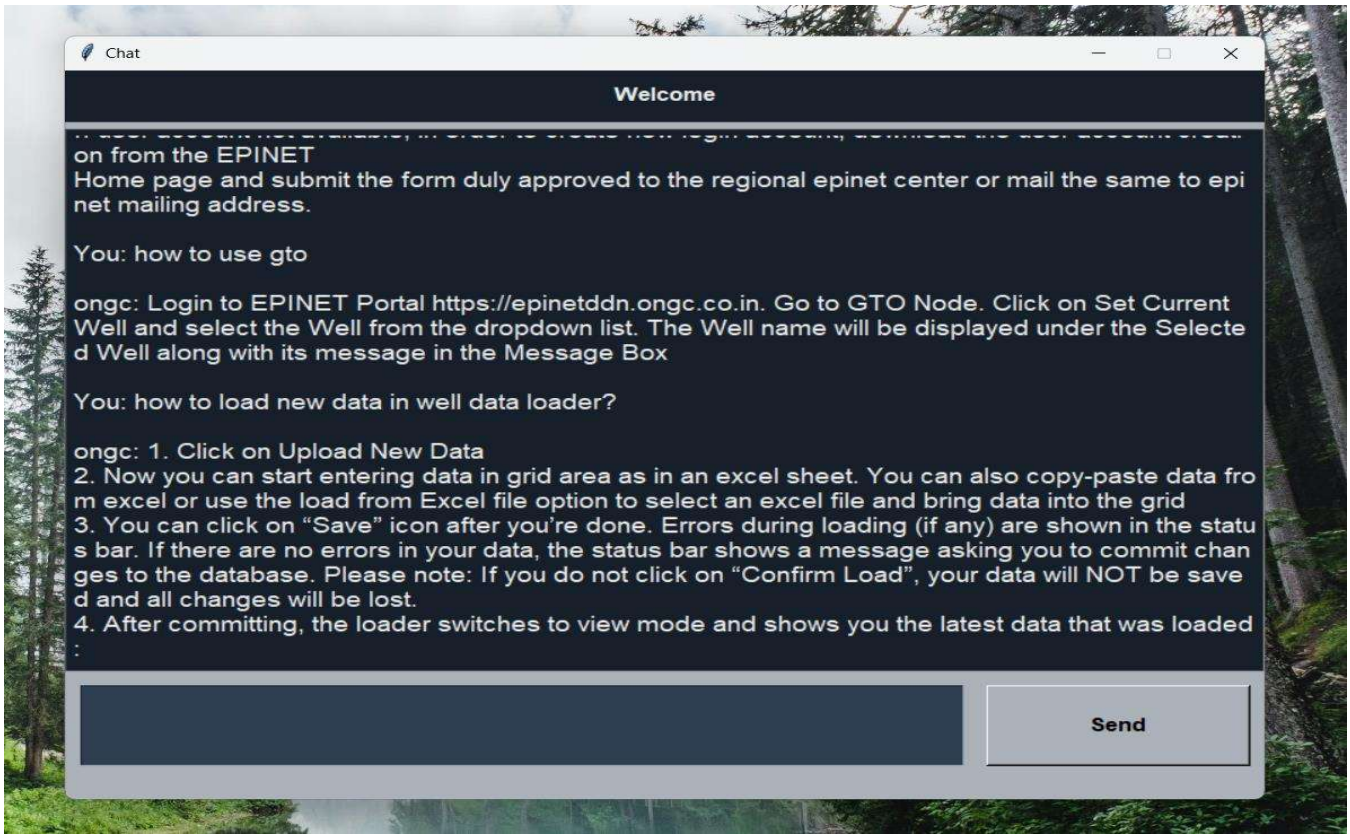
1. Firstly, execute the train.py script, it trains the Model with training data, Examines the loss and then saves the trained model's state and relevant metadata to a file for later use.



2. Now for running chatbot, run app.py script which has the script of GUI using Tkinter and uses chat.py script to implement the chat.

Here is ChatBot :





Project Scope

The scope of the ONGC Chatbot Project encompasses the development, deployment, and continuous improvement of a chatbot designed to facilitate communication and provide information within the Oil and Natural Gas Corporation (ONGC). The project aims to leverage natural language processing (NLP) and machine learning (ML) technologies to create an intelligent, responsive, and user-friendly virtual assistant. The following sections detail the key components and boundaries of the project scope:

1. Objective

- **Primary Goal:** To develop a chatbot that can handle common queries related to ONGC's operations, services, and general information.
- **Secondary Goals:** Enhance user engagement, improve response times, and reduce the workload on human support staff.

2. Functional Scope

- **User Interaction:** The chatbot will interact with users through a graphical user interface (GUI) built using Tkinter, allowing for easy and intuitive communication.
- **Query Handling:** The chatbot will be capable of understanding and responding to a predefined set of queries, which are categorized into various intents such as greetings, farewells, information about ONGC, and specific functionalities of EPINET.
- **Response Generation:** Using a trained neural network model, the chatbot will generate appropriate responses based on user inputs.

3. Technical Scope

- **Technologies Used:**
  - **Programming Languages:** Python.
  - **Libraries and Frameworks:** PyTorch for neural network implementation, NLTK for natural language processing, and Tkinter for GUI development.
- **Data Handling:** Use of JSON files to store and manage intents and responses. The chatbot will utilize a dataset containing various patterns and responses for training the neural network.
- **Machine Learning Model:** Development of a feedforward neural network model trained to classify user inputs into predefined categories and generate corresponding responses.
- **Model Training and Evaluation:** The model will be trained using supervised learning techniques, with performance evaluated through loss metrics during training.

4. Operational Scope

- **Deployment:** The chatbot application will be deployed as a standalone desktop application, accessible to ONGC employees and stakeholders.
- **Maintenance and Updates:** Regular updates will be implemented to expand the chatbot's knowledge base, improve its accuracy, and add new functionalities as needed.
- **User Feedback:** Mechanisms will be in place to gather user feedback, which will be used to refine and enhance the chatbot's performance over time.

5. Limitations

- **Predefined Responses:** The chatbot is limited to responding to queries for which it has been specifically trained. It may not handle unexpected or highly complex questions effectively.
- **Language and Context Understanding:** While the chatbot uses advanced NLP techniques, its ability to understand nuanced language and context may be limited compared to human operators.
- **Scope of Knowledge:** The chatbot's knowledge is restricted to the information contained in its training data and intents. It will not have real-time access to external data sources or databases.

6. Future Enhancements

- **Integration with Databases:** Future versions may integrate with ONGC's internal databases and systems for real-time data access and more comprehensive responses.
- **Advanced NLP Capabilities:** Implementing more sophisticated NLP algorithms and expanding the training dataset to improve the chatbot's understanding and response accuracy.
- **Multi-Platform Deployment:** Extending the chatbot's availability to mobile and web platforms to increase accessibility for users.

This project aims to deliver a functional and effective chatbot that enhances user experience and operational efficiency within ONGC. The scope covers the essential technical and functional aspects required to achieve this goal, while also acknowledging current limitations and potential areas for future improvement.

Challenges Faced

1. Data Collection and Preprocessing

- **Inconsistent Data**: The initial dataset contained inconsistencies in formatting. Normalizing and cleaning the data to ensure uniformity required significant effort.
- **Tokenization and Stemming**: Accurately tokenizing and stemming words in various patterns was challenging, especially for complex or domain-specific terms used within ONGC. Ensuring that the stemming process did not distort the meaning of these terms was crucial.

2. Building the Bag of Words Model

- **Handling Large Vocabulary**: With a diverse range of patterns, the vocabulary size grew large, leading to sparse and high-dimensional bag of words vectors. Balancing the vocabulary size while maintaining meaningful feature representation was a key challenge.
- **Word Ambiguity**: Some words appeared in multiple contexts with different meanings, making it difficult to accurately represent their significance in the bag of words model.

3. Model Training

- **Hyperparameter Tuning**: Finding the optimal set of hyperparameters (e.g., learning rate, batch size, number of epochs) was a time-consuming process. It required extensive experimentation and validation to achieve the best performance.
- **Overfitting**: The model sometimes overfitted the training data, leading to poor generalization on unseen data. Implementing regularization techniques and adjusting the model complexity helped mitigate this issue.
- **Training Time**: Training the neural network for 1000 epochs, especially on large datasets, was computationally intensive and time-consuming. Efficient use of hardware resources and managing long training times were critical.

4. Integration and Deployment

- **Compatibility Issues**: Ensuring compatibility between various libraries (e.g., PyTorch, NLTK, Tkinter) and handling version conflicts posed challenges during development and deployment.
- **Real-Time Performance**: Achieving real-time performance for user interactions required optimizing the model inference time and ensuring that the chatbot responded promptly.

5. User Interface Development

- **Designing an Intuitive UI**: Creating a user-friendly and intuitive interface with Tkinter required careful consideration of layout, responsiveness, and ease of use.
- **Handling User Inputs**: Managing and processing diverse user inputs robustly, including handling typos, slang, and varied query structures, was challenging.

6. Maintaining Accuracy and Relevance

- **Dynamic Knowledge Base**: Keeping the chatbot's responses accurate and up-to-date with ONGC's evolving information and knowledge base required continuous updates to the intents and responses.
- **Response Relevance**: Ensuring that the chatbot's responses were contextually relevant and helpful, especially for complex or ambiguous queries, was a significant challenge.

Despite these challenges, through iterative development and continuous improvement, the chatbot project successfully achieved its objectives, resulting in a robust and user-friendly solution for ONGC.

Future Enhancement

1. Natural Language Understanding (NLU) Improvements

- **Enhanced NLP Techniques**: Incorporate advanced NLP techniques such as transformers (e.g., BERT, GPT) to improve the chatbot's understanding of complex queries and context.
- **Entity Recognition**: Implement Named Entity Recognition (NER) to better understand and extract specific information from user queries, allowing more accurate and detailed responses.

2. Multilingual Support

- **Language Expansion**: Extend the chatbot's capabilities to support multiple languages, catering to a diverse user base. This can involve training models on multilingual datasets and implementing translation services.
- **Dialect and Regional Variants**: Include support for regional dialects and language variants, ensuring the chatbot is accessible and relevant to users from different regions.

3. Contextual Awareness

- **Contextual Memory**: Develop the ability for the chatbot to maintain context over a conversation, allowing it to provide more coherent and contextually appropriate responses.
- **Personalization**: Implement user-specific personalization features where the chatbot can remember user preferences, previous interactions, and tailor responses accordingly.

4. Integration with ONGC Systems

- **Seamless Integration**: Integrate the chatbot with ONGC's internal systems (e.g., databases, ERP systems) to provide real-time data access and support for complex queries related to operations, employee information, and more.
- **Automation of Tasks**: Enhance the chatbot's functionality to automate routine tasks such as scheduling meetings, booking facilities, and retrieving specific data points.

5. User Interface Enhancements

- **Rich Media Support**: Enable the chatbot to handle and respond with rich media content such as images, videos, and documents, providing a more engaging and informative user experience.
- **Voice Interaction**: Implement voice recognition and speech synthesis capabilities, allowing users to interact with the chatbot using natural spoken language.

6. Advanced Analytics and Reporting

- **User Interaction Analytics**: Develop advanced analytics to track user interactions, identify common queries, and gather insights into user behavior. This data can help refine the chatbot's responses and identify areas for improvement.
- **Feedback Mechanism**: Implement a feedback mechanism where users can rate responses and provide suggestions, helping to continuously improve the chatbot's performance and accuracy.

7. Enhanced Security and Compliance

- **Data Privacy**: Strengthen data privacy measures to ensure compliance with regulations such as GDPR, protecting user data and maintaining trust.
- **Security Protocols**: Implement advanced security protocols to safeguard against vulnerabilities and ensure the integrity of the chatbot system.

8. Scalability and Performance Optimization

- **Scalability**: Optimize the chatbot's architecture to handle increased user traffic and queries efficiently, ensuring consistent performance during peak usage times.
- **Cloud Deployment**: Consider deploying the chatbot on cloud platforms to leverage scalability, reliability, and advanced computational resources.

By implementing these future enhancements, the ONGC chatbot can continue to evolve, providing even more value to users and supporting the organization's goals with advanced, user-friendly, and efficient conversational capabilities.

Conclusion

The development of the ONGC chatbot marks a significant advancement in our efforts to enhance user engagement and streamline information retrieval within the organization. This project has successfully demonstrated the potential of artificial intelligence and natural language processing in creating intelligent, responsive, and user-friendly systems tailored to meet the specific needs of ONGC.

The chatbot's ability to handle a wide range of queries, provide accurate and contextually relevant responses, and integrate seamlessly with ONGC's internal systems showcases its versatility and robustness. Through the systematic approach of data collection, preprocessing, model training, and deployment, the project has laid a strong foundation for future enhancements and scalability.

Despite the challenges faced during the development process, such as data inconsistencies, model overfitting, and real-time performance optimization, the team's dedication and innovative solutions have led to the successful implementation of a reliable chatbot. The project has provided valuable insights into the intricacies of developing AI-driven systems and highlighted the importance of continuous improvement and adaptation to evolving user needs.

Looking ahead, the proposed future enhancements, including multilingual support, contextual awareness, advanced analytics, and improved integration with ONGC systems, will further elevate the chatbot's capabilities and utility. These enhancements will ensure that the chatbot remains a cutting-edge tool that not only meets but exceeds user expectations, fostering greater efficiency and satisfaction.

In conclusion, the ONGC chatbot project stands as a testament to the transformative power of technology in enhancing organizational processes. It serves as a crucial step towards a more connected, efficient, and user-centric future for ONGC. The continued development and refinement of this chatbot will undoubtedly contribute to the ongoing success and innovation within the organization.

References

1. https://chatbotsmagazine.com/contextual-chat-bots-with-tensorflow-4391749d0077

2. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. Retrieved from https://www.nltk.org/book/

3. Paszke, A., et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Retrieved from https://arxiv.org/pdf/1912.01703

4. Training Data ‘ 2020_EPINET_Manual_for_users.pdf ’ from ONGC.
.

5. Chatbot using Deep Learning and NLP. Retrieved from: https://ijarcce.com/wp-content/uploads/2022/03/IJARCCE.2022.11306.pdf