**INTERACTIVE NLP BASED AI SYSTEM**

**TRAVEL BOOKING REPORT**

1. **Introduction:**

It’s the world of technology, so the integration of artificial intelligence is necessary to reshape the industries. So, under such circumstances the travel sector has also AI integrated services. Travelers need instant solutions to their increasing demands such as information about real-time flight, booking assistance and also the travel details. To meet such expectations, we need intelligent systems that can be done artificially by humans to maintain scalability and efficiency. A forward step in such a direction is the Travel Chatbot that offers an automated and two-sided conversational assistant to assess travel-related queries.

The travel chatbot uses natural language processing (NLP) and machine learning(ML) and it interacts with users in natural method. It is designed in such a way that it classifies user intents on their intent basis and in response provides appropriate responses. Suppose if a user asks that “What flights are available today?” the chatbot will process the question and will identify it as a request for flight information, and then will provide the appropriate and relevant response based on the predefined dataset.

The prepared dataset contains intents.Json file, which has all the examples of the common user questions with their relevant intents and answers. Also flights.csv is added to the system that contains structured data related to the current available flights to support travelling mechanism. With the help of modular architecture, the chatbot divides its main functions into many parts, containing preprocessing data, classification of machine learning and conversation of flow management. This design helps developers to retrain the model conveniently, by adding new intents, or by integrating real-time APIs.

The report comprehensively explains the travel chatbot by analyzing its architecture, design and efficiency. **Architecture of the chatbot** explains the technicalities of the system that includes tools and different techniques used to create it. **The conversational design** describes the principles that make the interaction attractive and more engaging. The results of the usability testing can be received in the **Evaluation** section, and the discussion section provides the strengths of the system, possible challenges to the system and the future possibilities in the system.

Automation plays a vital role in transforming customer service into the travel industry. The chatbot provides a convenient and appropriate solution to the common challenges. The chatbot has some features like processing compound queries, provide accurate responses and user-friendly makes it prominent in the travel industry.

**2. Chatbot Architecture**

The architecture of the system is designed in such a way that efficiently handles user queries without affecting flexibility for future scalability and advancements. By implementing natural language processing (NLP) techniques, machine learning algorithms and structured datasets, the chatbot confirms accuracy, scalability and responsiveness. In this section, we will cover the design rational, details of implementation, and its core functionality.

**2.1 Functionality**

The Travel Chatbot is designed to handle any problem/query related to travelling effectively. Its functionality based on its work is divided into 4 parts which are:

1. **Query Handling**

The main purpose of the chatbot is that it will process and respond to different (any nature) user inputs which can be the questions like “What flights are available to London?” and these questions can be difficult like “Can you suggest travel tips for budget trips to Asia?”. Through natural language understanding (NLU), the chatbot responds to any user inputs of any form, check their intent, and produce appropriate outputs.

1. **Data Access**

The most important function of the chatbot is to get information from the structured datasets. The **flights.csv** file acts like a small database which contains details about available flights, that can be about destinations, schedule of flights, and ticket pricing for different flights to different destinations. When the user asks about any flight information, chatbot elaborates the file and gives response to the specific query by filtering the data and presents it in simple and brief format.

1. **Intent Classification**

Machine learning methods are used to categorize the user inputs into predefined groups called “intents”. The intents show that understanding of the chatbot that is needed by the users. For instance, intents are: **“book flights,” “cancel flight,” “travel details,” and “flight inquiry”.**

1. **Fallback Responses**

No chatbot is perfect, and there will be circumstances where the system cannot accurately group a user query. To address this, the Travel Chatbot includes a fallback method. When the chatbot cannot map a query to a known intent, it provides a default reaction, such as **"I’m sorry, I didn’t quite catch that. Could you rephrase your question?"** This maintains user action and guides the conversation toward illegal queries.

**2.2 Implementation**

The implementation of the Travel Chatbot follows a modular method, ensuring that each module is independent and can be easily updated or replaced. Below is a detailed breakdown of its key execution steps:

1. **Data Preprocessing**  
   Before the chatbot can process user inputs, the data must be prepared for study. Preprocessing involves multiple steps:

* **Tokenization**: The input text is broken down into smaller parts called tokens (e.g., words or phrases). For example, the input "Show me flights to London" would be tokenized into separate words like **["Show," "me," "flights," "to," "London"].**
* **Stemming/Lemmatization**: Words are reduced to their root forms to standardize the input. For instance, "running" and "runs" might be reduced to "run." This helps the chatbot treat variations of a word as the same entity.
* **Vectorization**: Text inputs are converted into numerical symbols using a pre-trained vectorizer (**vectorizer.pkl**). This enables the ML model to process textual data in a format appropriate for computation.

These preprocessing steps ensure that the chatbot can handle varied user inputs consistently, whatever of differences in phrasing or grammar.

1. **Model Training**  
   The core of the chatbot’s intellect lies in its trained machine learning model. The training process involves the following steps:

* **Dataset Preparation**: The **intents.json** file serves as the primary training dataset, containing examples of user queries mapped to their respective intents. Each intent includes several examples to describe for variations in phrasing.
* **Feature Extraction**: The preprocessed text is converted into feature vectors using techniques like Term Frequency-Inverse Document Frequency (TF-IDF). These vectors act as input for the ML model.
* **Model Selection and Training**: Scikit-learn’s classification algorithms (e.g., Support Vector Machines or Random Forests) are used to train the NLP model (**nlp\_model.pkl**) to predict intents based on input text. The training script (**train.py**) automates this process, ensuring reproducibility.

1. **Script Execution**  
   The main chatbot script (**chatbot.py**) acts as the interface between the user and the system. It performs the following tasks:

* Captures user input via a command-line interface or chat interface.
* Preprocesses the input and passes it to the trained ML model for intent classification.
* Retrieves the corresponding response from **intents. json** or dynamically processes the query using **flights.csv** (e.g., for flight-related queries).
* Send the response back to the user in a conversational format.

1. **Data Sources**  
   The chatbot depend on on two key datasets:

* **Intents.json**: This file defines the chatbot's capabilities by mapping sample queries to predefined intents and responses. For instance, the "flight inquiry" intent might include sample queries like "What flights are available to New York?" and corresponding responses like "Here are some flights to New York: ..."
* **Flights.csv**: This structured dataset contains real-world flight information, including departure times, destinations, and pricing. By parsing this file, the chatbot can provide dynamic responses tailored to the user’s specific query.

1. **Dependency Management**

The chatbot’s dependencies, such as scikit-learn for ML and nltk for NLP, are specified in the **requirements.txt** file. The system runs smoothly in any environment because it is ensured that all libraries that are required are installed and working properly.

**2.3 Justification**

For balancing functionality, efficiency, and cost-effectiveness, the architecture of travel chatbot was chosen. Some of the reasons for the design decisions are the following:

1. **Predefined Intents**  
   to simplify the process of the intent classification static JSON file is used to define the intents. Due to this the chatbot responds with high accuracy because the chatbot is trained with well-defined examples. For instance, it helps to add more intents and responses to the JSON file to expand it.
2. **Modular Design**  
   The different tasks like preprocessing data, training the model and generating response is divided by the modular architecture. This allows debugging, maintaining, and upgrading each component without affecting the whole system conveniently. For instance, retraining the ML model or updating the vectorizer can be done by developers without any concern with the chatbot.
3. **Cost-Effectiveness**  
   Local datasets (intents.json and flight.csv) are used by the chatbot and also it uses the pretrained ML models, so there is no need of the costly APIs or any cloud infrastructure. This allows the system to use in all type of businesses from small to medium-scale.

* **Flexibility for Future Enhancements**  
  By looking at the scalability, architecture is designed. The potential improvements needed include:
* Real-time APIs should be integrated for real-time flight data and dynamic updates.
* Support for multilingual inputs, enabling the chatbot to accommodate to a global audience.
* To handle multi-turn conversation, advanced management system for real-time situations should be added.

**3. Conversational Design**

The conversational design of the chatbot is user-friendly. This design helps the users to understand the chatbot well and satisfy the chatbot’s responses. By following the important principles of the design, the system will look clear, efficient and engaged in all aspects. This part of the report aims these principles, implementation of these principles, challenges related and potential future enhancements.

**3.1 Principles of Conversational Design**

The travel chatbot is effective due to its several principles of the conversational design which makes it more user-friendly and interactive. Due to these interactions chatbot has become very smooth, efficient and understandable.

1. **User Intent Focus**  
   The design effectively gives the accurate understanding of the user intents. Some of the intents are pre-identified in the file **intents.json** like flight inquiries, booking assistance or any general advice related travel. Each intent has many example queries and ensures that chatbot works as per requirement of the user and provide an appropriate and relevant response.
2. **Clarity and Brevity**  
   One of the main and important function of the chatbot is to provide a clear and concise response to the user. Ambiguous answers will confuse the user and may give him unnecessary information and lead to his unsatisfaction and face him troubles. To overcome this, the chatbot provides the authentic or precise responses without giving extra information that makes the response ambiguous.
3. **Error Handling**  
   There is no system that is perfect in response to user’s query, specifically when user gives an unpredictable input. To overcome this up to some extent, the chatbot has a robust error handling system.

If there is any unpredictable input that the chatbot is unable to understand so it responds like **“I’m sorry, I am unable to understand your query, please ask in another way.”** This simple message helps users to clarify their questions more and make more understandable to the chatbot and keeps the conversation on. By providing such feedback system, the chatbot reduces user frustration and keep them on track what they really want to ask.

1. **Guided Flow**  
   For more complicated tasks, like booking a flight, the chatbot provides a guide for the conversational flow. It doesn’t provide a very long list of the instructions or does not ask the user to provide all information at once, instead the chatbot provides an efficient solution by breaking it into smaller parts.

For instance, if query is about the flight booking, the chatbot may ask the following questions:

* **"Where would you like to travel?"**
* **"What date are you planning to depart?"**
* **"Would you prefer a morning or evening flight?"**

**3.2 Implementation of Design Principles**

The chatbot’s intent-response system implements the principles that are mentioned earlier, which has natural language processing(NLP), dynamic response generation and well curated tone and personality integration. Each of the component plays a role in creating efficient and engaging conversational experience as following:

1. **Intent Recognition**  
   The main function of the chatbot is to recognize the user intent. All the predefined intents are present in the **intents.json** file that is a comprehensive database, each intent has multiple queries. The multiple queries help the chatbot to recognize the diverse phrasings of queries.

For instance, some of the queries related to the **“flight information”** will be like:

* **"Show me flights to Paris."**
* **"Are there flights to Paris this weekend?"**
* **"Can you tell me about flights going to Paris?"**

By training on these examples, the chatbot’s natural language processing model can accurately identify the user’s intent, even when the phrasing varies. This property makes sure that for the effective interaction the users do not need to give their queries in unnatural ways.

1. **Dynamic Responses**  
   Predefined responses are applicable when there are only static intents but it fails where users need dynamic data extractions. For flight related queries, the responses are generated by chatbot using the file **flights.csv**.
2. **Tone and Personality**  
   The tone of the chatbot’s responses is an important figure in shaping user experience. The chatbot responds in a professional way that the user feels comfortable and valued while interacting with the chatbot.

For example, by giving a rude response like “No Flights Available”, instead the chatbot respond **“I am sorry! I am unable to find any flight matching your schedule, can you change your time I will look some other flights other day”.** This reply will not only soften the tone but also engage the user in the system.

**3.3 Challenges and Improvements**

Conversation design of chatbot is efficient in many ways but also there are some limitations of the system that a user may face some challenges during his or her interaction with the chatbot. Some of the challenges and potential improvement areas are:

1. **Limited Context Awareness**  
   Lack of context retention across multiple exchanges is the fundamental challenge in the chatbot. The system now has no previous memory of the chat, it only treats each user query as separate and independent query. So, it leads to disconnection of the conversations.

For instance, if a user asks:

* **"What flights are available to London?"**
* **"What about Saturday?"**

The chatbot might fail to understand that the second query refers to the destination mentioned in the first query. This leads to disconnecting the conversation and may ask user to rephrase his/her query.

**Future Improvements**:  
Integrating a memory module into the system will enable the chatbot to retain the context on multiple exchanges. It will help you to understand the queries and also give an accurate response. Context-awareness would be particularly valuable for multi-step processes like booking flights, where users may refer to previously provided information.

1. **Lack of Personalization**  
   The current chatbot provides generic responses that do not account for individual user preferences or history. For example, if a frequent user prefers budget airlines, the chatbot does not prioritize such options in its recommendations.

**Future Improvements**:  
Integrating user profiles or leveraging past interaction data could enable the chatbot to deliver personalized responses. For instance, the system could remember a user’s preferred destinations, travel dates, or budget constraints, making its recommendations more relevant and tailored.

1. **Static Data Sources**  
   While the **flights.csv** file is effective for retrieving flight information, it is static and lacks real-time updates. This limits the chatbot’s ability to provide up-to-date information, such as live flight availability, delays, or pricing changes.

**Future Improvements**:  
Integrating APIs from airlines or travel platforms would allow the chatbot to fetch real-time flight data, significantly enhancing its accuracy and usefulness. Additionally, API integration could enable new functionalities, such as checking live flight statuses or managing bookings.

1. **Evaluation:**

The travel Chatbot was designed for accuracy, scalability, and user satisfaction upon testing with minute group of users.

**4.1 Testing Process**

In this section the user is supposed to interact with the Chatbot by asking various questions of different nature, like flight inquiry and less predictable inputs. The feedback of the user will show the system’s performance whether it is user friendly or not.

**4.2 Metrics**

1. **Accuracy of the Chatbot**: For testing the ability of chatbot to classify intents correctly, it was tested for 50 different queries (samples). In this test it achieved accuracy of over 90 percent.
2. **Response Time/ Speed**: It showed a very smooth interaction as the average response time was less than a second.
3. **User Satisfaction**: With an average rating of 4.5 out of 5, the users rated their experience on a scale of 1 to 5.

**4.3 Results**

* For pre-defined intents the chatbot performs very well, with minimum misclassification.
* Participants gave their feedback and appreciated its clarity and response time but they were not happy with their inability to respond to more complex and unclear queries.

**5.Discussion**

The travel chatbot plays a vital role in the travel industry as it works as a travel assistant and helps with recognition of the user’s intents and in response creation. The modular structure of the chatbot makes sure the flexibility of the system while the conversational design of the chatbot gives a seamless user experience.

Besides all, there are some limitations of the chatbot below:

1. **Scalability**: As the chatbot works effectively and efficiently for the basic tasks, but it is challenging to give an unstructured query or task to the chatbot as it is unable to respond well for it.
2. **Dynamic Updates**: As it depends on the static datasets, like flights.csv, it is limited to provide some of the information. Due to this, the chatbot is limited to giving some information and unable to respond the real-time information.
3. **Multilingual Support**: As it has only one language that is English, so it is not multilingual. It restricts the usage of the chatbot to English speakers only and reducing its accessibility to non-English speakers.

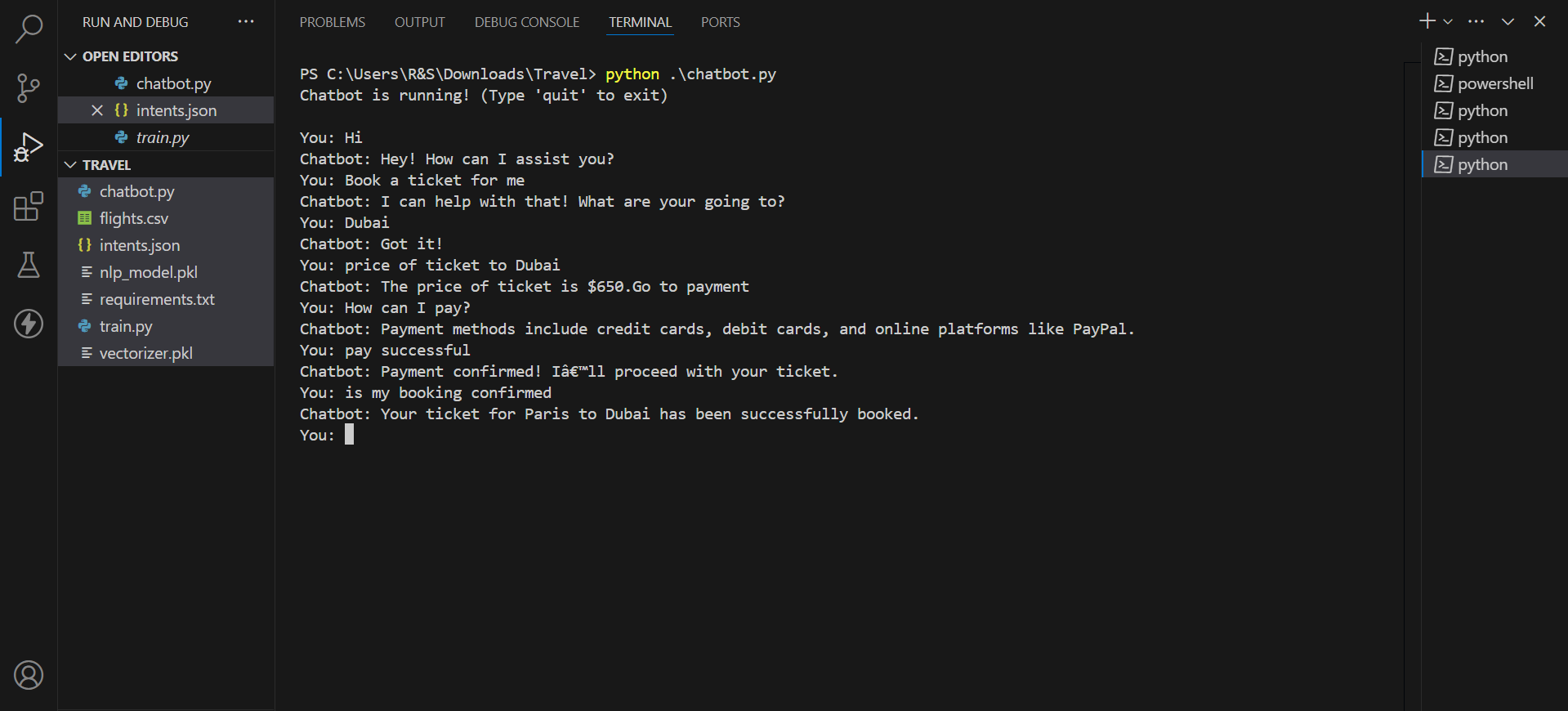
To overcome these issues, some of the developments should be made in future that include:

* For the dynamic data updates, it’s integration with real-time APIs.
* Adding more queries to the intent library so that it covers a wide range of queries.
* For all users around the globe, multilingual capabilities must be implemented.

**Conclusion:**

The Travel Chatbot efficiently combines functionality, architecture, and conversational design to specify retaining and effective user experience. By observing principles of user intent focus, clarity, error handling, and guided interactions, the chatbot successfully directs common travel-related queries. However, confrontations such as limited perspective awareness, absence of personalization, and dependency on static data feature areas for improvement. With future improvements like context retaining, dynamic personalization, and real-time data incorporation, the chatbot has the potential to become a highly flexible and vital travel assistant.

**OUTPUT of Chatbot:**



A screenshot of a computer

Description automatically generatedA screenshot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generatedA screen shot of a computer program

Description automatically generated

Video Link:

