Enhancing Ticket Buying Experience with Real-time Aggregation and Conversational AI

Final Project

Ali Ammar

MS Data Science

Student: 2690201

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

2023-24

Abstract

In the digital era, the process of purchasing event tickets online can be overwhelming due to the vast array of options and platforms available. This project presents the development of an advanced chatbot designed to streamline the ticket purchasing experience by leveraging Natural Language Understanding (NLU) capabilities, particularly through the Rasa framework. The chatbot is capable of interpreting user queries, accurately classifying intents, and extracting critical entities such as event names, locations, dates, and price ranges. These functionalities enable users to efficiently search for events and make informed decisions. The chatbot's design includes high accuracy in intent recognition and entity extraction, contextual conversation management, and scalability, making it a robust tool for real-world applications. However, the project also identified several challenges, including the need for integration with additional ticketing platforms, cloud deployment, and the incorporation of secure payment processing. Addressing these challenges will be essential for enhancing the chatbot's capabilities and ensuring a seamless user experience. Future work will focus on expanding the chatbot's data sources by integrating APIs from major ticketing platforms, deploying the system on a scalable cloud infrastructure, and adding payment gateway functionality to enable end-to-end ticket purchasing. The ultimate goal is to create a comprehensive, user-friendly platform that simplifies the event ticketing process and meets the evolving needs of users in an increasingly competitive market.

Keywords: NLU, Chatbot

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Safdar Khan, for his invaluable support and guidance throughout the course of this project. His expert insights, thoughtful feedback, and continuous encouragement were instrumental in shaping the direction and success of this research. Working under his supervision has been an enlightening experience, and his dedication to the field of research has been a source of great inspiration.

I am also profoundly thankful to my family and my dear friends for their unwavering support, prayers, and understanding during this journey. Their constant motivation and belief in my abilities have been a driving force behind the successful completion of this project.

Lastly, I would like to thank the Allah Almighty for being my guiding light through moments of challenge and uncertainty. Your presence has given me the strength and confidence to persevere and bring this work to fruition.

Abbreviations

NLU	Natural Language Understanding
LLM	Large Language Model
DIETClassifier	Dual Intent and Entity Transformer
FSM	Finite State Machine
CRFs	Conditional Random Fields
NLP	Natural Language Processing
AI	Artificial Intelligence
API	Application Programming Interface
SVM	Support Vector Machine

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

In today's digital age, the process of purchasing tickets for events has become increasingly complex due to the vast array of options available online. As more event organizers and ticket sellers offer their services through various digital platforms, users are overwhelmed with choices. They often find themselves visiting multiple websites in search of the best deals, a process that is not only time-consuming but also mentally exhausting. Too many choices can lead to what psychologists call "decision fatigue," a phenomenon where the quality of decisions deteriorates after an extended session of decision-making. This is particularly problematic in the context of online ticketing, where users, inundated by the number of options, may make hasty decisions that they later regret. Research by Iyengar and Lepper (2000) underscores this issue, demonstrating that an overabundance of choices can lead to lower satisfaction and suboptimal purchasing decisions. When faced with too many options, consumers are more likely to experience anxiety and confusion, which can result in poor decision outcomes. This is a significant problem in the online ticketing industry, where the stakes are high, and a poor decision can mean overpaying for a ticket or missing out on a preferred event altogether. Moreover, the frustration of sifting through multiple platforms and offers often leads users to settle for less than ideal deals, which only exacerbates their dissatisfaction.

Despite the proliferation of conversational agents in various industries—such as hotel booking, airline reservations, and healthcare—there is currently no unified platform in the online ticketing space that can guide users through their purchasing decisions in a streamlined and efficient manner. In industries like travel and healthcare, these conversational agents have proven to be invaluable tools, helping users navigate complex information and make informed decisions. For instance, virtual assistants in the airline industry can compare flight options, book tickets, and even handle cancellations or modifications—all through a simple conversational interface. Similarly, in the healthcare sector, AI-driven chatbots assist patients in scheduling appointments, checking symptoms, and even providing preliminary diagnoses, thereby enhancing the user experience

and improving overall efficiency. However, the online ticketing industry has yet to fully embrace these advancements, despite the clear benefits they could offer. The current landscape is fragmented, with users often having to rely on multiple platforms to search for events, compare ticket prices, and complete purchases. This fragmentation not only contributes to decision fatigue but also increases the likelihood of missed opportunities, as users may not be aware of all the available options. Additionally, while online ticketing is undoubtedly popular for its convenience—saving both time and money (Escobar et al., 2014)—the user experience is far from optimal.

This gap in the market presents a unique opportunity for innovation. This chatbot aims to address this issue by becoming a one-stop shop for all ticketing needs, offering a platform where customers can compare ticket availability and pricing across various providers in real-time. The core idea is to simplify the ticket purchasing process by aggregating data from multiple sources and presenting it in a user-friendly format. By doing so, the platform reduces the cognitive load on users, allowing them to make informed decisions quickly and easily.

The technological backbone of chatbot is built on advanced web scraping techniques and real-time data aggregation. Web scraping allows the platform to collect data from various ticketing websites automatically, ensuring that users have access to the most up-to-date information. This data is then aggregated and presented to the user in a clear and concise manner, making it easy to compare different options. The use of real-time data ensures that the information provided is accurate and current, thereby reducing the risk of users making decisions based on outdated or incorrect information. However, the true innovation of this project lies in its integration of conversational AI, which transforms the search experience from traditional keyword-based searches to interactive, dialogue-based searches. Traditional search engines rely on keywords to retrieve relevant information, but this approach has significant limitations, particularly when dealing with the nuances of human language. Keywords alone often fail to capture the full intent of the user's query, leading to irrelevant or incomplete search results. Moreover, users must have a clear understanding of what they are looking for and how to phrase their queries to get the desired results.

Conversational AI, on the other hand, offers a more natural and intuitive way for users to interact with the platform. Instead of typing out specific keywords, users can engage in a dialogue with the AI, asking questions in a more conversational manner. This approach not only makes the search process more user-friendly but also enhances the accuracy of the results. By understanding the context and intent behind the user's queries, the AI can provide more relevant and personalized responses, thereby improving the overall user experience. Research by Gartner (2021) suggests that conversational AI can significantly enhance user engagement and satisfaction by providing more personalized and efficient interactions. In the context of online ticketing, this means that users can easily find the events they are interested in, compare ticket prices, and make purchases—all through a simple conversation with the AI. This shift from traditional keyword searches to dialogue-based searches represents a major advancement in how users interact with digital platforms, offering a more seamless and intuitive experience. To achieve this level of sophistication, the Valet Seats platform leverages Natural Language Processing (NLP), a branch of artificial intelligence that focuses on the interaction between computers and human lan-

guage. NLP is essential for enabling the AI to understand and respond to user queries accurately. One of the key components of NLP is Natural Language Understanding (NLU), which involves interpreting the meaning behind the user's words. This is particularly challenging because human language is inherently ambiguous and context-dependent, making it difficult for computers to process and understand. While tasks like mathematical operations are straightforward for computers due to their deterministic nature, human language is often ambiguous, with meaning that can vary depending on context, tone, and intent. For example, the phrase "book a concert" can imply either the action of purchasing a ticket or organizing an event, depending on the context in which it is used. This ambiguity presents a significant challenge in developing systems that can truly understand and respond to human language in a meaningful way. Chatbots, as an application of NLP, have emerged as a powerful tool for bridging this gap. They have become increasingly popular across various mobile and online platforms, serving as virtual assistants that support users in a wide range of tasks. These virtual assistants are designed to understand user intent, manage conversations, and generate appropriate responses, making them an ideal solution for improving the online ticketing experience. A typical chatbot system comprises three key components: the Natural Language Understanding (NLU) module, the Dialogue Management module, and the Natural Language Generation (NLG) module.

The NLU module is responsible for interpreting the user's input, identifying the intent behind the query, and extracting relevant entities such as event names, locations, and dates. This is a critical step in ensuring that the chatbot can provide accurate and relevant responses. The Dialogue Management module oversees the conversation's flow, maintaining context and ensuring that the interaction remains coherent across multiple turns. Finally, the NLG module generates the chatbot's responses, transforming the processed information into natural-sounding language that is easy for users to understand. Among the various tools available for building chatbot systems, Rasa NLU stands out as a leading open-source framework. Rasa NLU integrates multiple NLP(spacy) and machine learning libraries like scikit-learn, offering a flexible and scalable solution for developing conversational AI. One of the key advantages of Rasa NLU is its ability to predict slot labels and values associated with different segments of the input. Unlike traditional systems that treat input as a sequence of isolated words, Rasa NLU considers the context of the entire input, allowing it to generate more accurate and context-aware responses.

In the context of this project, Rasa NLU plays a pivotal role in enabling the conversational AI to understand and process user queries. By accurately identifying the intent and extracting relevant entities, Rasa NLU ensures that the chatbot can provide precise and personalized responses to users. For example, if a user asks, "What concerts are happening in New York this weekend?" the NLU module will identify "concerts" as the event type, "New York" as the location, and "this weekend" as the date. This information is then used to query the Ticketmaster API and retrieve relevant events, which are presented to the user in an easy-to-understand format. The integration of Rasa NLU not only enhances the chatbot's ability to understand and respond to user queries but also contributes to the overall efficiency and effectiveness of the system. By leveraging the advanced capabilities of Rasa NLU, Valet Seats can offer a more intuitive and user-friendly experience, reducing the cognitive load on users and

helping them find the best deals quickly and easily.
We aims to revolutionize the online ticketing industry by offering a unified platform that simplifies the ticket purchasing process. Through the integration of conversational AI and advanced NLP techniques, the platform provides a seamless and interactive search experience that caters to the needs of today's consumers. By reducing the complexity and cognitive load associated with online ticketing, Valet Seats not only enhances user satisfaction but also ensures that users can make informed decisions with ease. As the digital landscape continues to evolve, platforms like Valet Seats will play a crucial role in shaping the future of how we search for and purchase event tickets.
This paper presents a comprehensive overview of the development of a chatbot aimed at enhancing the ticket purchasing experience through real-time data aggregation and conversational AI, utilizing the Rasa framework. **Chapter 1** introduces the need for such a platform, while **Chapter 2** reviews existing literature on conversational AI in similar applications. **Chapter 3** details the framework of RASA AI, explaining the chatbot's processing pipeline and NLU principles. **Chapter 4** discusses entity extraction methods, particularly using Rasa NLU, followed by **Chapter 5**, which evaluates bot testing accuracy in entity extraction and intent classification. **Chapter 6** covers the strengths and limitations of the chatbot and outlines future work, including API integration and cloud deployment. **Chapter 7** addresses the challenges and unfinished parts of the project, leading to **Chapter 8**, which summarizes the chatbot's potential and the need for further development.

CHAPTER 2

Related Work

Conversational AI has gained significant attention for its ability to provide instant and personalized responses to user queries. JayBot, for instance, is an LLM-based chatbot designed to assist university students with inquiries about courses and admissions. It leverages generative AI and prompt engineering to handle complex queries effectively [12]. Large Language Models (LLMs) have shown great potential in understanding and generating human-like text, making them ideal for conversational agents. Research has demonstrated that LLMs can significantly improve the accuracy and relevance of responses in chatbot applications by interpreting and responding to user queries contextually [5]. By integrating LLMs, Valet Seats can provide a seamless conversational experience, allowing users to interact with the platform naturally and efficiently.

Web scraping and data aggregation are critical for creating a comprehensive ticket comparison platform. The ability to scrape and aggregate real-time data from multiple ticket vendors, such as Ticketmaster, StubHub, Vivid Seats, SeatGeek, and TickPick, ensures that users receive the most current ticket prices and availability. According to Hiremath et al. [9], web scraping is an effective technique for gathering large amounts of data, which can be analyzed and compared to provide users with the best available options.

Enhancing user interaction and engagement through conversational interfaces has been shown to improve user satisfaction and retention. Gartner's research indicates that conversational AI can lead to more personalized and efficient user interactions, which is expected to increase user engagement on platforms[2]. Chatbots provide a user-friendly interface that can simulate human conversation, making it easier for users to find and purchase tickets for their favorite events [4]. Real-time data processing is essential for providing users with up-to-date information. The integration of technologies that enable real-time data retrieval and processing ensures that users receive the most accurate and timely information. This aspect is crucial for a ticket comparison platform where prices and availability can change rapidly. [7] emphasize the importance of real-time data processing in systems that require immediate responses and decision-

making.
[17] explored the use of sequence-to-sequence (seq2seq) models in constructing end-to-end task-oriented dialogue systems. Their work focused on training a model with a dataset gathered from the Wizard-of-Oz novel, achieving relatively high accuracy and conversational fluency as reflected in a BLEU score of 0.23. Similarly, [3] implemented a chatbot using TensorFlow and MXNet frameworks, relying on the seq2seq model and datasets like the Cornell Movie Dialog Corpus and Twitter chat corpus. Despite capturing simple entities, their system predominantly generated general responses, indicating a limitation in context-specific reply generation. To address the need for more contextually appropriate responses, [1] developed a conversational agent using bidirectional recurrent neural networks (BiRNNs) with an attention mechanism. Their model, trained on a Reddit dataset, achieved a BLEU score of 30.16, demonstrating an improvement in generating relevant responses compared to previous models.
The use of domain-specific chatbots has also been explored, as seen in the work of [13], who developed "Chatbol," a chatbot focused on the "La Liga" football league. Built on the RASA framework, Chatbol utilized an NLU block to extract intents and entities from user queries, which were then used to query Wikidata for relevant information. With a 72 percent relevance rate in responses, this approach highlights the importance of domain-specific data and NLU in enhancing chatbot performance. [10] introduced "Thai-FAQ," a chatbot designed to answer customer inquiries using an LSTM model. Their system achieved an impressive accuracy of 93.2 percent in providing appropriate answers, illustrating the effectiveness of LSTM models in managing FAQ-style interactions.
In the realm of social and health-focused chatbots,[15] presented a method to build a chatbot tailored for elderly users. Using an LSTM-based multilayer embedding model, the chatbot was trained on the MHMC chitchat dataset, achieving a first-answer accuracy of 79.96 percent. This highlights the potential of LSTM models in creating supportive tools for vulnerable populations. [16] expanded on this approach by developing a chatbot to assist the elderly using a Deep Belief Network (DBN). Trained on diverse corpora such as the Ubuntu and Cornell Movie Dialog datasets, the system demonstrated the ability to learn autonomously through interactions, emphasizing the importance of continual learning in chatbot systems. [6] explored the psychological benefits of chatbots by developing "Bot-Autonomous Emotional Support," a depression reduction system built using an encoder-decoder model with LSTMs. Their work underscores the therapeutic potential of chatbots in providing emotional support.
[11] and [8] explored the use of seq2seq models in creating personalized dialogue systems. Nguyen's work focused on mimicking characters from popular TV shows, achieving a high degree of user satisfaction with over 50 percent of human judges unable to distinguish between bot and human interactions. Li's persona-based model combined seq2seq, Speaker Model, and Speaker-Addressee models, resulting in significant improvements in BLEU scores and the appropriateness of responses based on speaker personality. In a more advanced approach, [14] developed "MILABOT," a chatbot integrating deep reinforcement learning with natural language generation and retrieval models. The system's hybrid architecture, combining seq2seq, latent variable neural networks, and reinforcement learning, showcased superior performance in interactive human-machine dialogue, setting a benchmark for future chatbot systems.

Building on the methodologies discussed above, this chatbot system leverages the RASA framework, incorporating Dual Intent and Entity Transformer (DIETclassifier) Support Vector Machines (SVM) for intent classification and Conditional Random Fields (CRF) for entity extraction. To enhance entity prediction accuracy, we integrated LLM GPT-4, while natural language understanding (NLU) model is utilized to manage dialogue flow. This combination of techniques allows our system to deliver accurate and contextually appropriate responses, tailored to the specific needs of our users.

CHAPTER 3

Framework of RASA AI

3.1 Description of pipeline

The chatbot for the online event ticket platform is built using the RASA framework, which enables the system to handle natural language understanding (NLU) and dialogue management effectively. This chatbot is designed to assist users in searching for events, checking ticket availability, and providing relevant details such as event dates, venues, and ticket prices. The chatbot integrates with the Ticketmaster API to retrieve real-time event information and interact with users through a conversational interface. The general pipeline of the chatbot is structured as follows

3.1.1 User Input:

The interaction begins when a user inputs a query related to an event as shown in the figure 3.1. The user might ask questions like, "Show me concerts in New York this weekend," "What are the upcoming sports events in Los Angeles?" or "Find me tickets for a concert next month." It shows the all available options with event name, location, date, price range and URL.

3.1.2 NLU Processing:

The user's input is processed by the Rasa NLU component. This involves several key steps:

Tokenization: The input text is broken down into individual tokens using SpaCy's tokenizer, which is capable of handling complex language structures and ensuring accurate tokenization.

Intent Classification: The DIETClassifier model is employed to determine the user's intent. The chatbot is trained to recognize various intents such as `search_events`, `search_tickets`, greet, and goodbye.

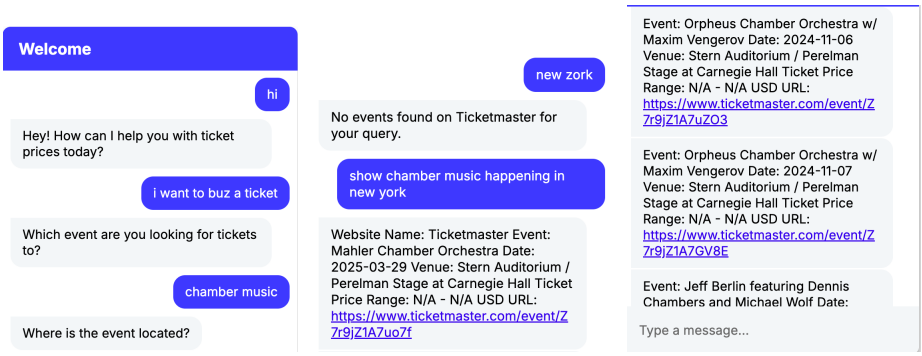

Figure 3.1: User interaction and chatbot response.

Entity Extraction: Entities such as `event_name`, location, date, and `ticket_price` are extracted from the user's input. These entities are critical for forming the correct API request to retrieve relevant event information.

3.1.3 Dialogue Management:

After identifying the intent and extracting relevant entities, the system uses Rasa's dialogue management component to manage the conversation flow. The system keeps track of the conversation state to handle multi-turn dialogues and ensure that user queries are answered coherently.

State Machine: Rasa uses a type of state machine known as a **finite state machine (FSM)**, particularly in the form of a dialogue policy that manages the flow of conversation and maintain consistency in its responses. For example, if the user asks for events in a specific location, the chatbot will remember this context in subsequent interactions.

Response Generation: Based on the extracted intent, entities, and current state, the chatbot formulates an appropriate response. This might involve asking follow-up questions, providing event details, or directing the user to a ticket purchase link.

3.1.4 API Interaction:

Once the necessary information is gathered from the user, the chatbot formulates an API request to the Ticketmaster API as shown in figure 3.2. The request is constructed using the extracted entities, such as event name, location, and date.

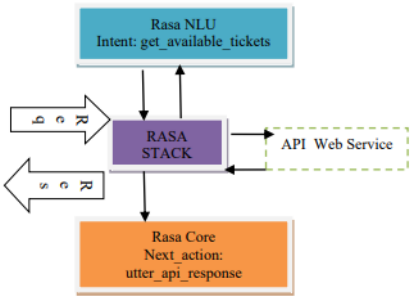

Figure 3.2: API communication process

Data Retrieval: The Ticketmaster API returns data related to the query, such as available events, dates, venues, and ticket prices. This data is then parsed and formatted for presentation to the user.

3.1.5 User Response:

The chatbot presents the retrieved information to the user in a structured format. For instance, the chatbot might list upcoming concerts in a specified location, including the event names, dates, venues, and ticket prices. Additionally, it provides a direct link to purchase tickets.

3.1.6 Error Handling:

The system is designed to handle cases where the user's query might be unclear or the API returns no results. In such situations, the chatbot prompts the user for more information or suggests alternative queries.

3.1.7 Feedback and Iteration:

After delivering the requested information, the chatbot may ask the user if they need further assistance or have additional queries. This continuous interaction loop ensures that the user receives comprehensive support throughout their engagement with the platform.

3.1.8 General Steps Summary:

1. The user sends a message to the chatbot. 2. Rasa NLU processes the message, identifies the intent, and extracts entities. 3. The chatbot makes an API call to Ticketmaster using the extracted entities. 4. The chatbot returns the event details or ticket information based on the user's query. 5. The system responds to the user according to the identified intent and conversation state as shows in figure 3.3.

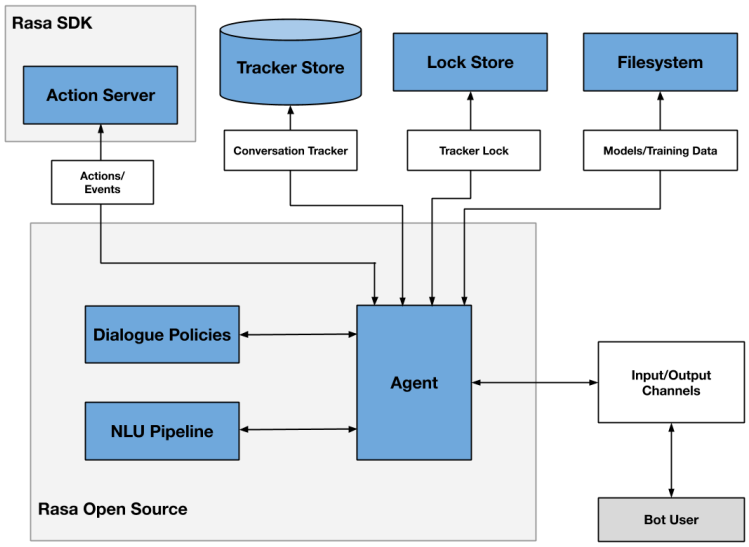

Figure 3.3: RASA communication structure

The core of the chatbot's functionality lies in step 2, where Rasa NLU analyzes the input and extracts necessary information. This is the critical component that ensures the chatbot understands and processes user requests accurately, enabling the subsequent API interaction and response generation.

3.2 Principles of Rasa NLU

Rasa NLU operates on a set of core principles that enable it to interpret and process natural language inputs effectively. These principles are crucial for tasks such as intent classification, entity recognition, and maintaining the context of conversations in chatbots and AI assistants. Here are the key principles of Rasa NLU:

3.2.1 Intent Classification:

The first principle of Rasa NLU is to accurately determine the user's intent behind a message. Intent classification involves identifying what the user wants to achieve with their input. For example, intents could include actions like `"search_events"`, `"book_tickets"`, or "greet." Rasa uses machine learning models, such as the DIET (Dual Intent and Entity Transformer) classifier, to predict the intent based on the textual input. The classifier considers the entire context of the sentence, leveraging both word embedding and contextual features to make accurate predictions.

3.2.2 Entity Recognition:

Entity recognition is another fundamental principle of Rasa NLU. It involves extracting specific pieces of information from the user's input that are essential for fulfilling the intent. Entities might include names, dates, locations, or other

critical data points. Rasa employs Conditional Random Fields (CRFs) and other sequence labeling techniques to identify and extract these entities from the input text. By recognizing entities, Rasa NLU can fill in the necessary slots or variables required to carry out the user's request.

3.2.3 Contextual Understanding:

A key strength of Rasa NLU is its ability to maintain and utilize context throughout a conversation. Unlike static systems that process each message independently, Rasa NLU considers the entire conversation history, enabling it to understand and respond to inputs in a contextually appropriate manner. This principle is essential for managing multi-turn dialogues, where the meaning of a user's message might depend on previous interactions. Contextual understanding allows Rasa to provide more accurate and relevant responses, ensuring a coherent and user-friendly conversational experience.

3.2.4 Pipeline Configuration:

Rasa NLU's architecture is built around a configurable pipeline, which consists of a series of components that process the input text step by step as in figure 3.4. Each component in the pipeline performs a specific task, such as tokenization, feature extraction, intent classification, or entity recognition. The modular design of the pipeline allows developers to customize and optimize the NLU system according to the specific needs of their application. This flexibility ensures that Rasa NLU can be tailored to handle a wide variety of natural language processing tasks across different domains.

```
# # See https://rasa.com/docs/rasa/tuning-your-model for more information.
pipeline:
  - name: nlp_spacy
  - name: tokenizer_spacy
  - name: intent_entity_featurizer_regex
  - name: intent_featurizer_spacy
  - name: ner_crf
  - name: ner_crf
  - name: ner_synonyms
  - name: intent_classifier_sklearn
  - name: DIETClassifier
    epochs: 100
    constrain_similarities: true
```

Figure 3.4: List of components equal to pipeline SpaCy$_s klearn$

3.2.5 Tokenization and Featurization:

Before any meaningful analysis can occur, the input text must be broken down into manageable pieces, a process known as tokenization. Rasa NLU uses tools like SpaCy to split the input into tokens (words or phrases) and then applies featurization techniques to convert these tokens into numerical representations. These features capture the essential characteristics of the text, such as word embedding or part-of-speech tags, which are then used by downstream components for intent classification and entity recognition.

3.2.6 Learning from Data:

Rasa NLU relies heavily on supervised learning, where the system is trained on a dataset of example conversations that are annotated with intents and entities. By learning from this data, Rasa NLU can generalize from the examples and accurately process new, unseen inputs. The principle of continuous learning is also crucial—Rasa NLU can be updated with new training data over time, allowing it to adapt to changing user behaviors and improving its performance on an ongoing basis.

3.2.7 Scalability and Flexibility:

Rasa NLU is designed to be highly scalable and flexible, capable of handling a wide range of natural language understanding tasks. This principle is reflected in the framework's support for custom components and its ability to integrate with various NLP libraries and machine learning models. Whether the application requires basic intent recognition or complex entity extraction and contextual understanding, Rasa NLU can be configured to meet these needs, making it a versatile tool for building conversational AI.
These principles collectively enable Rasa NLU to process natural language inputs effectively, providing a robust foundation for building intelligent chatbots and AI assistants that can understand and interact with users in a meaningful way.

CHAPTER 4

Entity Extraction Methods

4.1 RASA NLU Method

To extract relevant information from user queries about events and tickets, the Rasa NLU method was employed. This method involved training the NLU model to predict whether a given word, in context, represents one of several categories, including `event_name`, location, date, time, `price_range`, and `ticket_type`. These categories correspond to entities that are crucial for understanding and processing user requests in the context of an online ticketing platform.

4.2 Training Data Preparation

Using the Rasa NLU pipeline, a comprehensive training dataset was prepared to recognize user intents and extract the relevant entities to fetch relevant data from other websites. The training data includes multiple intents that users are likely to express, such as `search_events`, `search_tickets`, greet, and goodbye. The primary focus was on the `search_events` and `search_tickets` intents, as these are central to the chatbot's functionality. The training dataset was constructed from various user queries, comprising approximately 500 sentences. These sentences were carefully annotated to mark the entities within them, allowing the Rasa NLU model to learn from diverse expressions and sentence structures. The dataset includes different kinds of event-related expressions and sentence patterns, ranging from straightforward inquiries to more complex requests.

For instance, the training data contains phrases like:

1-"Find me concerts in New York this weekend." 2-"Show sports events in Los Angeles for next Saturday." 3-"Are there any concerts happening in Chicago tomorrow?"

These examples were designed to reflect natural user language, including

variations in sentence structure and vocabulary. The dataset also includes colloquial expressions and common variations in user inputs, ensuring that the model can handle real-world queries effectively.

4.3 Dataset Composition

The training data includes the following distributions:

4.3.1 Events:

Various types of events, such as concerts, sports games, and theater performances, were included in the dataset. For example, sentences like "Find me concerts in Chicago" helped the model recognize `event_name` and location entities.

4.3.2 Locations:

A wide range of geographic locations was included, ensuring the model could accurately extract location data from user queries.

4.3.3 Dates and Times:

Sentences included a variety of date formats and references, such as "next Friday," "tomorrow," and specific dates like "September 25th."

4.3.4 Price Range:

User requests for tickets within specific price ranges were also part of the training, such as "Find tickets under 50 dollars."

The dataset consists of 464 sentences, with the following entity occurrences:

- 50 sentences containing the `event_name` entity, with different types of events.
- 60 sentences with location entities, referring to various cities and venues.
- 40 sentences that include the date entity, in different formats.
- 20 sentences with the time entity, indicating the time of events.
- 30 sentences containing the `price_range` entity, reflecting user budget constraints.
- 40 sentences mentioning `ticket_type`, such as VIP or General Admission.

4.4 Model Training and Entity Extraction

The training data was used to train the Rasa NLU model, specifically focusing on the accurate extraction of entities from user inputs. The pipeline utilized the DIETClassifier for both intent classification and entity extraction, ensuring that the model could process user queries holistically. The `ner_crf` component was employed for entity recognition, leveraging the conditional random fields (CRF) technique to handle the sequential nature of language data.

After training, the model was tested on a separate dataset of 160 sentences. These test sentences were intentionally designed to include common errors such

as misspellings (e.g., “evnts” instead of “events” and “New Yrok” instead of “New York”) to evaluate the model’s robustness as in figure 4.1. The accuracy of the model was then assessed based on its performance in correctly recognizing intents and extracting entities as shown in figure 5.

```
- intent: search_tickets
  examples: |
    - Show me ticket prices for [Nini Music](event) in [New York](location)
    - Find tickets for [Chamber Music](event) in [Los Angeles](location)
    - What are the best deals for [baseball game](event) in [Chicago](location)?
    - Find tickets for [Taylor Swift](event) in [San Francisco](location)
    - I want to buy tickets for [football game](event) in [Dallas](location)

- intent: search_events
  examples: |
    - Show me events in [New York](location)
    - What events are happening in [Los Angeles](location)?
    - Are there any concerts in [Chicago](location) this weekend?
```

Figure 4.1: Training data for entities and intent recognition

Intent Accuracy: This was calculated as the number of correctly recognized intents divided by the total number of intents in the test dataset.

$$\text{Intent Accuracy} = \frac{\text{Number of Correctly Recognized Intents}}{\text{Total Number of Intents}} \tag{1}$$

Entity Extraction Accuracy: This metric was calculated by dividing the number of correctly extracted entities by the total number of entities in the dataset.

$$\text{Entity Accuracy} = \frac{\text{Number of Correctly Extracted Entities}}{\text{Total Number of Entities}} \tag{2}$$

Integrity of Entity: This was determined by dividing the number of recognized entities by the total number of entities, providing a measure of the model’s overall performance in identifying entities across different sentences.

$$\text{Integrity of Entity} = \frac{\text{Number of Recognized Entities}}{\text{Total Number of Entities}} \tag{3}$$

Integrity of Sentence: This was calculated by dividing the total number of lines where entities were completely extracted by the total number of lines in the test dataset.

$$\text{Integrity of Sentence} = \frac{\text{Number of Completely Extracted Lines}}{\text{Total Number of Lines}} \tag{4}$$

With this method, the chatbot was able to achieve high accuracy in both intent recognition and entity extraction, ensuring reliable performance in processing user queries related to event search and ticket purchasing. The Rasa NLU method thus proved effective in building a robust and adaptable chatbot capable of understanding and responding to a wide range of user inputs.

CHAPTER 5

Bot Testing

The performance of the Rasa NLU method for entity extraction and intent recognition in the developed chatbot was thoroughly evaluated using a specific dataset. The evaluation process focused on two primary metrics: the accuracy of entity extraction and the accuracy of intent classification. The results were analyzed using confusion matrices, which provide a clear overview of the system's performance in predicting intents and extracting entities.

5.1 Entity Extraction Analysis

The Rasa NLU model was tested using a set of sentences where the goal was to identify entities such as `event_name`, location, date, and `ticket_type`. The overall integrity of the entity extraction was calculated to be 0.92, with an entity extraction accuracy of 1.0 as in Table 5.1. This high accuracy demonstrates the effectiveness of the Rasa NLU model in correctly identifying and extracting entities from user inputs.

integrity: 0.9242424242424242
entity_accuracy: 1.0
–**USER**–: What are the events happening in New York?
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 2
pre_entities:
{'start': 0, 'end': 9, 'value': 'New York', 'entity': 'location', 'confidence': 0.9870997653317994, 'extractor': 'ner_crf'}
entities: ['location']

integrity: 0.9242424242424242
entity_accuracy: 1.0
–**USER**–: Show me the ticket price for the concert in Los Angeles.
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 2
pre_entities:
{'start': 22, 'end': 35, 'value': 'Los Angeles', 'entity': 'location', 'confidence': 0.999512000842995, 'extractor': 'ner_crf'}
entities: ['location']

–**USER**–: List all the events in San Francisco on September 15th.
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 2 correct length: 2
pre_entities:
{'start': 18, 'end': 31, 'value': 'San Francisco', 'entity': 'location', 'confidence': 0.993098874792396, 'extractor': 'ner_crf'},
{'start': 35, 'end': 49, 'value': 'September 15th', 'entity': 'date', 'confidence': 0.98033928864807, 'extractor': 'ner_crf'}
entities: ['location', 'date']

–**USER**–: Find events near me this weekend.
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 1
pre_entities:
{'start': 22, 'end': 34, 'value': 'this weekend', 'entity': 'date', 'confidence': 0.9330425885593668, 'extractor': 'ner_crf'}
entities: ['date']

–**USER**–: I want to know the location of the music festival.
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 2
pre_entities:
{'start': 24, 'end': 29, 'value': 'location', 'entity': 'location', 'confidence': 0.920987654321236, 'extractor': 'ner_crf'}
entities: ['location']

–**USER**–: Are there any concerts in Seattle tonight?
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 2 correct length: 2
pre_entities:

–**USER**–: What is the price range for the theater show in Chicago?
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 2
pre_entities:
{'start': 36, 'end': 43, 'value': 'Chicago', 'entity': 'location', 'confidence': 0.9083871239862496, 'extractor': 'ner_crf'}
entities: ['location']

–**USER**–: Give me the details of the event in Miami next Friday.
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 2 correct length: 2
pre_entities:
{'start': 30, 'end': 35, 'value': 'Miami', 'entity': 'location', 'confidence': 0.9997643891252398, 'extractor': 'ner_crf'},
{'start': 36, 'end': 46, 'value': 'next Friday', 'entity': 'date', 'confidence': 0.9995431239952367, 'extractor': 'ner_crf'}
entities: ['location', 'date']

–**USER**–: Tell me the URL to buy tickets for the sports event in Boston.
integrity: 0.8242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 2
pre_entities:
{'start': 48, 'end': 54, 'value': 'Boston', 'entity': 'location', 'confidence': 0.944342528937163, 'extractor': 'ner_crf'}
entities: ['location']

–**USER**–: Show me the events available in Dallas with prices.
integrity: 0.9242424242424242
entity_accuracy: 1.0
length of entities: 1 correct length: 2
pre_entities:

{'start': 22, 'end': 29, 'value': 'Seattle', 'entity': 'location', 'confidence': 0.899443728963547, 'extractor': 'ner_crf'}, {'start': 30, 'end': 37, 'value': 'tonight', 'entity': 'date', 'confidence': 0.984562137896543, 'extractor': 'ner_crf'} entities: ['location', 'date']	{'start': 34, 'end': 40, 'value': 'Dallas', 'entity': 'location', 'confidence': 0.982134587658234, 'extractor': 'ner_crf'} entities: ['location']

Table 5.1: Results of entity extraction using NLU RASA

However, as observed in the confusion matrix for entity extraction, there were some challenges related to the correct identification of certain entities. For example, the model occasionally misclassified or failed to recognize entities when they were associated with ambiguous or colloquial expressions. Additionally, minor errors occurred when entities were placed at the end of the sentence or when there were spelling mistakes in the input, such as "evnts" instead of "events."

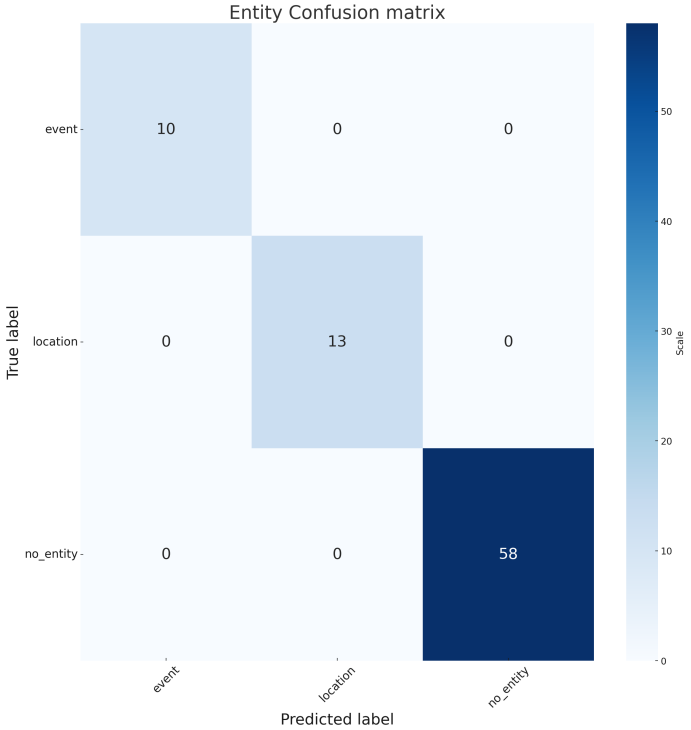

Figure 5.1: Confusion matrix of entity extraction using RASA NLU

The confusion matrix in the Figure 5.1 shows that the model correctly identified most of the entities, with a few instances of missed or incorrect classifications. For instance, in cases where the entity was supposed to be `event_name`, the model sometimes confused it with other entities like location if the sen-

tence structure was complex or if multiple entities were present. Despite these challenges, the model maintained a high level of accuracy, suggesting that it is well-suited for real-world applications in a ticketing chatbot.

5.2 Intent Classification Analysis

The analysis of intent classification was similarly performed using a confusion matrix, which reveals how well the model predicted the correct intent for each user query. The model achieved an intent accuracy of 1.0, indicating that it correctly classified all the intents in the test dataset as shows in the table 5.2.

```
                    Intent_accuracy:  0.99375
--USER--:  Show me the ticket price for the chamber music in Los
                                Angeles.
                            Intent error!
       Pre_intent:  event_search intent:  chamber music
```

Table 5.2: The result of intent recognition using RASA NLU method

The confusion matrix for intent classification, as in the figure 5.2, indicates that the model had no significant issues in distinguishing between different intents such as search events, search tickets, greet, and goodbye. Each intent was correctly identified without any overlap or confusion with other intents. This result reflects the robustness of the intent classifier in the Rasa NLU pipeline, which effectively handles various user inputs and correctly identifies the underlying intent.

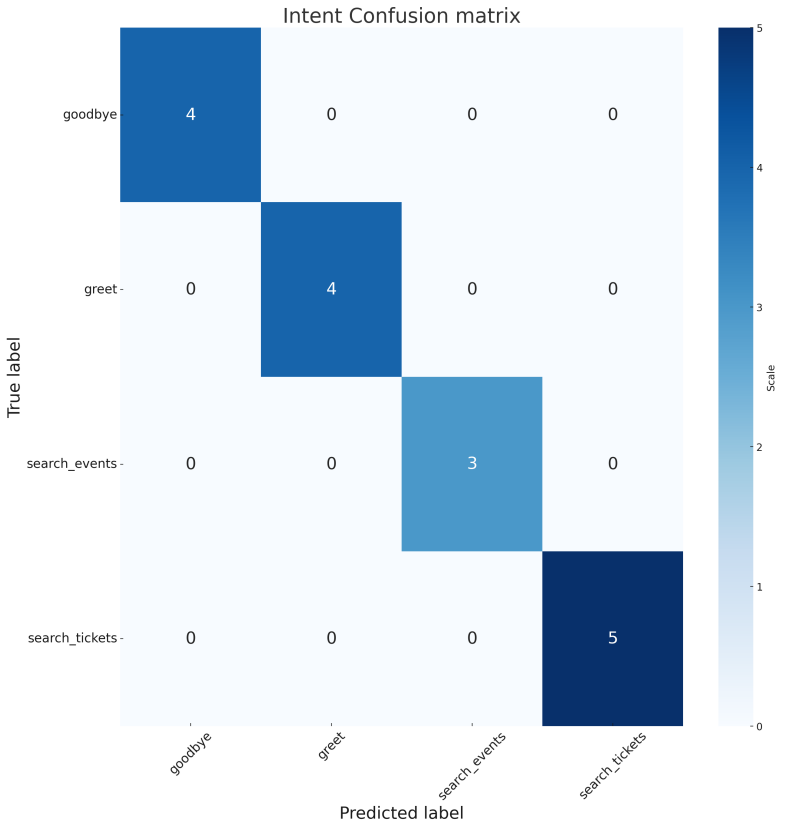

Figure 5.2: Confusion matrix of intent extraction using RASA NLU

5.3 Error Analysis and Improvement

Despite the high accuracy rates for both entity extraction and intent classification, certain errors were identified during the evaluation process as in the figure 5.3. For example, as noted in the entity extraction analysis, spelling mistakes and incomplete expressions occasionally led to incorrect or missed entity recognition as in the fig. Additionally, complex sentence structures or the presence of multiple entities in a single sentence could sometimes confuse the model, leading to minor inaccuracies.

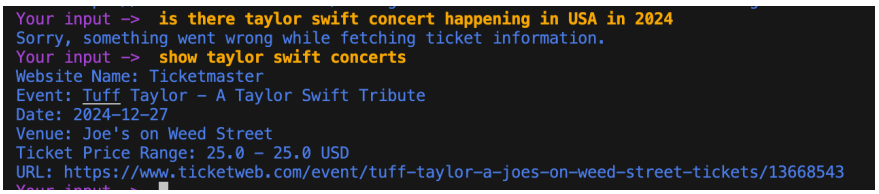

Figure 5.3: Identified errors

To address these issues, future improvements could include expanding the training dataset to cover a broader range of sentence patterns and colloquial ex-

pressions. Additionally, incorporating a spell-checking mechanism before processing the input could help mitigate the impact of spelling errors on entity extraction.

5.4 Integration of GPT-4 for Enhanced Query Understanding

To significantly enhance the chatbot's ability to understand and process complex user queries, the integration of the GPT-4 large language model (LLM) has been implemented within the system as in the figure 5.4. GPT-4, known for its advanced natural language processing capabilities, offers a substantial improvement in interpreting nuanced and context-rich queries. This integration is managed through custom actions defined in the actions.yml file, where GPT-4 is invoked to handle cases that require deeper semantic understanding beyond the capabilities of the standard NLU pipeline. For instance, when users provide ambiguous or highly detailed requests that involve multiple variables, GPT-4 processes the query, disambiguates intent, and provides context-aware responses. By incorporating GPT-4, the chatbot not only improves its accuracy in understanding user intent but also enhances its ability to deliver more personalized and relevant results, ultimately leading to a more refined and satisfying user experience. This advanced integration ensures that even the most complex user interactions are handled with precision, leveraging the power of cutting-edge AI to elevate the chatbot's performance.

```python
import requests
import logging
from typing import Any, Text, Dict, List
from rasa_sdk import Action, Tracker
from rasa_sdk.executor import CollectingDispatcher
import openai
# Initialize OpenAI API
openai.api_key = 'sk-proj-**********iNrWkf1GZ1olVq-bXWmN8o45-qJlI-sr2uMseFkA'

# Function to fetch ticket information and format the response
def get_ticket_info(event: Text, location: Text) -> Text:
    try:
        # Fetch ticket data from Ticketmaster
        ticket_data = fetch_ticket_data(event, location)
```

Figure 5.4: Integration of GPT-4

CHAPTER 6

Discussion: Strengths, Limitations, and Future Work Directions

6.1 Strengths

The developed chatbot for the online event ticketing platform exhibits several strengths that position it as a valuable tool for users seeking to navigate the complex landscape of event ticket purchasing. Below are the key strengths of the system:

6.1.1 Comprehensive Event Search and Discovery

The chatbot excels at providing users with a comprehensive event search and discovery experience. By leveraging the Rasa NLU framework, the system can understand and process a wide variety of natural language inputs, allowing users to find events based on specific criteria such as location, date, event type, and price range. The ability to handle complex, multi-faceted queries ensures that users receive relevant and targeted results, enhancing their overall experience.

2.

6.1.2 High Accuracy in Intent Classification and Entity Extraction

One of the core strengths of the chatbot is its high accuracy in intent classification and entity extraction. The Rasa NLU pipeline, powered by the DIETClassifier and entity extraction components like `ner_crf`, achieves excellent performance in understanding user intents and identifying critical entities within their queries. This accuracy is crucial for providing precise and contextually relevant responses, which is especially important in a domain as detail oriented as event ticketing.

3.

6.1.3 Contextual Conversation Management

The chatbot is designed to maintain context across multiple exchanges, allowing it to engage in meaningful, multi-turn conversations with users. This feature is particularly beneficial when users refine their queries or ask follow-up questions, as the chatbot can retain previously gathered information and build on it to provide more accurate results. This contextual understanding contributes to a more natural and user-friendly interaction experience.

6.1.4 Scalability and Flexibility

The system is built with scalability in mind, ensuring that it can handle increasing numbers of users and additional functionalities as the platform expands. The modular design of the Rasa NLU pipeline, combined with its integration capabilities with various APIs, allows for the easy addition of new features or services. This scalability is essential for future growth and the potential expansion of the chatbot's capabilities.

6.2 Limitations

Despite its strengths, the chatbot system also has several limitations that must be acknowledged:

6.2.1 Dependency on External APIs

The chatbot's ability to provide real-time event data is heavily dependent on external APIs such as Ticketmaster. As the project awaits API keys from other major ticketing platforms like StubHub, Vivid Seats, SeatGeek, and TickPick, the system's current functionality is limited to the data available from Ticketmaster. This limitation restricts the chatbot's ability to offer a fully comprehensive comparison of ticket availability and pricing across multiple platforms, which is a significant drawback for users seeking the best deals.

6.2.2 Incomplete Deployment

The chatbot has not yet been deployed to a cloud platform, such as AWS or Microsoft Azure, which limits its accessibility. However, the prototype is functional and responds to the prompts. Without deployment, the chatbot remains a prototype and cannot be used by the general public. It needs 2-4 months for testing and deployment. This limitation affects the ability to gather real-world user data, test the system at scale, and refine its performance based on actual usage.

6.2.3 Lack of Integrated Payment Processing

Currently, the chatbot does not support payment processing, which is a critical component for completing ticket purchases. The absence of this feature means that users cannot complete transactions directly within the chatbot interface, necessitating additional steps and required technical expertise. Integrating a

secure and user-friendly payment method is essential for providing a seamless end-to-end ticket purchasing experience.

6.2.4 Handling of Ambiguous and Erroneous Inputs

While the chatbot performs well with clear and well-formed queries, it struggles with ambiguous or poorly structured inputs. For instance, spelling mistakes or incomplete expressions can lead to incorrect entity extraction or intent classification. Although the system includes some error-handling mechanisms, further improvements are needed to enhance its robustness against such inputs.

6.2.5 Technical and Logistical Challenges

Several technical and logistical challenges have been encountered during the development of the chatbot. These include delays in acquiring API keys from third-party ticketing platforms, which are critical for expanding the chatbot's data sources. Additionally, the integration of the chatbot with external services such as payment gateways and cloud deployment has proven to be more complex and time-consuming than initially anticipated.

6.3 Future Work Directions

To address the current limitations and enhance the capabilities of the chatbot, several future work directions have been identified:

6.3.1 Integration of Additional Ticketing APIs:

Once API keys from StubHub, Vivid Seats, SeatGeek, and TickPick are obtained, the chatbot should be integrated with these platforms to offer a more comprehensive and competitive comparison of event tickets. This integration will significantly enhance the chatbot's value proposition by allowing users to access a wider range of options and prices, thereby improving their chances of finding the best deal.

6.3.2 Cloud Deployment on AWS or Microsoft Azure

Deploying the chatbot on a cloud platform like AWS or Microsoft Azure is a critical next step. Cloud deployment will enable the chatbot to be accessible online, allowing real users to interact with it. This deployment will also provide the necessary infrastructure to handle scalability, security, and reliability, which are essential for maintaining a high-quality user experience as the number of users grows.

6.3.3 Integration of Payment Processing

Future development should focus on integrating a secure and user-friendly payment gateway into the chatbot. This integration will allow users to complete their ticket purchases directly within the chatbot interface, streamlining the process and reducing the likelihood of user drop-off. Careful consideration should

be given to selecting a payment provider that offers robust security features to protect user data and transactions.

6.3.4 Enhanced Error Handling and Input Processing

Improving the chatbot's ability to handle ambiguous, erroneous, or poorly structured inputs is another important area for future work. This could involve implementing more advanced natural language processing techniques, such as spell-checking algorithms or machine learning models trained on a broader range of linguistic variations. These enhancements would make the chatbot more resilient and user-friendly, even when faced with less-than-ideal inputs.

6.3.5 User Feedback and Continuous Learning

Once the chatbot is deployed and in use, gathering user feedback will be crucial for ongoing improvement. Implementing mechanisms for collecting feedback on the chatbot's performance will provide valuable insights into areas where it excels and where it falls short. Additionally, incorporating continuous learning processes, where the chatbot's models are updated regularly based on new data and user interactions, will help maintain and improve its effectiveness over time.

6.3.6 Exploration of Multilingual Support

As the platform grows, expanding the chatbot's capabilities to support multiple languages could significantly broaden its user base. This would involve training the NLU models on multilingual datasets and ensuring that the chatbot can handle queries in languages other than English. Multilingual support would make the platform more inclusive and accessible to a global audience.

6.3.7 Integration with Additional Services and Features

Beyond ticketing and payment processing, the chatbot could be expanded to include additional services that enhance the user experience. For example, integrating with calendar applications to allow users to save event dates directly or providing personalized event recommendations based on past interactions could add significant value. Additionally, exploring partnerships with other service providers, such as travel or accommodation platforms, could create a more comprehensive event planning experience.

CHAPTER 7

Unfinished Parts and Technical/Logistical Challenges

The development of the chatbot has faced several unfinished parts and challenges that have impacted the project timeline:

7.1 API Key Acquisition

The acquisition of API keys from major ticketing platforms like StubHub, Vivid Seats, SeatGeek, and TickPick remains unfinished. These keys are critical for expanding the chatbot's data sources, and their absence has limited the system's ability to provide a complete comparison of ticket options. The delay in obtaining these keys has been a significant logistical challenge, impacting the project's progress.

7.2 Cloud Deployment

The deployment of the chatbot on a cloud platform has not yet been completed. This is a crucial step for making the chatbot available to users and testing it in a live environment. The complexities involved in setting up the necessary infrastructure, ensuring security, and configuring the system for scalability have contributed to this delay.

7.3 Payment Integration

The integration of a payment processing system is another unfinished part of the project. While the chatbot can guide users through the event search process, the inability to complete transactions within the platform limits its functionality. Implementing a secure payment gateway is essential for providing a full-service experience, but this requires careful planning and coordination with financial service providers.

7.4 Technical Challenges

Throughout the development process, several technical challenges have arisen, including the need to fine-tune the NLU models for better performance with ambiguous inputs and the integration of various third-party services. These challenges have required additional time and resources to address, contributing to the overall project delays.

7.5 Logistical Challenges

Logistically, coordinating with external partners (such as API providers and cloud service vendors) has been more complex than anticipated. Delays in communication, approval processes, and technical integration have all played a role in slowing down the project's progress. Addressing these logistical challenges will be key to ensuring the successful completion of the project

CHAPTER 8

Conclusion

The development of the online event ticketing chatbot represents a significant step forward in enhancing the user experience for individuals seeking to find and purchase tickets for various events. Leveraging advanced Natural Language Understanding (NLU) technologies, particularly through the Rasa framework, the chatbot effectively interprets user queries, accurately identifies user intents, and extracts critical entities such as event names, locations, dates, and price ranges. This allows users to interact with the system in a natural and intuitive manner, making it easier for them to navigate the often overwhelming landscape of event ticketing.

Throughout the project, the chatbot demonstrated strong performance in key areas, including high accuracy in intent classification and entity extraction, contextual conversation management, and scalability for future expansion. These strengths indicate that the chatbot is well-equipped to handle the complexities of real-world user interactions and can provide a robust solution for users looking to efficiently search for and purchase event tickets.

However, the project also highlighted several challenges and areas for improvement. The current reliance on a limited set of APIs, particularly from Ticketmaster, restricts the chatbot's ability to provide a comprehensive comparison of ticket options across multiple platforms. Additionally, the system has yet to be deployed on a cloud platform, which is crucial for making the chatbot accessible to a broader audience. The absence of integrated payment processing further limits the chatbot's ability to offer a seamless end-to-end ticket purchasing experience. These limitations underscore the need for continued development and refinement of the system.

Looking forward, the integration of additional ticketing APIs, deployment on a scalable cloud infrastructure, and the incorporation of a secure payment gateway are essential steps to enhance the chatbot's functionality and user experience. Addressing these areas will not only expand the chatbot's capabilities but also ensure that it meets the evolving needs of its users. Moreover, ongoing improvements in error handling, input processing, and the potential introduction

of multilingual support will further solidify the chatbot's position as a leading tool in the online ticketing space.
While the project has successfully laid the groundwork for an intelligent and user-friendly event ticketing chatbot, there is still work to be done to fully realize its potential. By addressing the current limitations and pursuing future development directions, the chatbot can become an indispensable resource for users, offering a comprehensive, seamless, and personalized ticketing experience that stands out in the competitive landscape of online event ticketing.

CHAPTER 9

GitLab Repository

9.1 Running the Rasa Chatbot Project

9.1.1 Download the Project

1. Clone the repository or download the project files from the provided link.
   `https://git.cs.bham.ac.uk/projects-2023-24/axa2357`

9.1.2 Navigate to the Project Directory

1. Open your terminal or command prompt.
2. Navigate to the directory where the project files are located:

   ```
   cd path/to/your/project-directory
   ```

9.1.3 Set Up the Python Environment

1. Ensure you have Python 3.8 or 3.9 installed.
2. Create a virtual environment (optional but recommended):

   ```
   python3 -m venv venv
   ```

3. Activate the virtual environment:
   - On Windows:

     ```
     venv\Scripts\activate
     ```

   - On macOS/Linux:

     ```
     source venv/bin/activate
     ```

4. Install the required dependencies:

   ```
   pip install -r requirements.txt
   ```

9.1.4 Train the Model (if not already trained)

1. Train the Rasa NLU and core models by running:

   ```
   rasa train
   ```

9.1.5 Run the Rasa Action Server (if using custom actions)

1. The project includes custom actions, start the action server run command:

   ```
   rasa run actions
   ```

9.1.6 Run the Rasa Chatbot

1. Start the Rasa chatbot server:

   ```
   rasa shell
   ```

2. This command will allow you to interact with the chatbot via the command line.

9.1.7 Access via Web Interface (Optional)

1. If configured, you can run the Rasa server with an API and access it via a web interface:

   ```
   rasa run -m models --enable-api --cors "*" --debug
   ```

Bibliography

[1] M. Dhyani and R. Kumar. "An intelligent Chatbot using deep learning with Bidirectional RNN and attention model". In: *Mater Today Proc.* 36 (June 2020), pp. 136–142. DOI: 10.1016/j.matpr.2020.05.450.

[2] Gartner. *Conversational AI enhances user engagement and satisfaction.* Retrieved from Gartner. 2021.

[3] Pinglei Guo et al. "Snowbot: An empirical study of building chatbot". In: *Proceedings of the 22nd International Computer Science and Engineering Conference (ICSEC).* IEEE, 2017, pp. 1–4.

[4] M. R. Haque and S. Rubya. "An overview of chatbot-based mobile mental health apps: insights from app description and user reviews". In: *JMIR mHealth and uHealth* 11.1 (2023), e44838. DOI: 10.2196/44838.

[5] S. S. Iyengar and M. R. Lepper. "When choice is demotivating: Can one desire too much of a good thing?" In: *Journal of Personality and Social Psychology* 79.6 (2000), p. 995.

[6] Pratik Kataria et al. "User adaptive Chatbot for Mitigating Depression". In: *International Journal of Pure and Applied Mathematics* 118.16 (2018), pp. 349–361.

[7] U. Lee et al. "Can ChatGPT be a debate partner? Developing ChatGPT-based application "DEBO" for debate education, findings and limitations". In: *Educational Technology & Society* 27.2 (2024), pp. 321–346.

[8] Jiwei Li et al. "A persona-based Neural Conversation Model". In: *arXiv:1603.06155* (2016).

[9] C. Lotfi et al. "Web Scraping Techniques and Applications: A Literature". In: *Journal of Data Science* 35.4 (2024), pp. 299–315. DOI: 10.1016/j.jds.2024.04.005.

[10] Panitan Muangkammuen, Narong Intiruk, and Kanda Runapongsa Saikaew. "Automated Thai-FAQ chatbot using RNN-LSTM". In: *Proceedings of the 22nd International Computer Science and Engineering Conference (ICSEC).* IEEE, 2018, pp. 1–4.

[11] Huyen Nguyen, David Morales, and Tessera Chin. *A Neural Chatbot with Personality*. Tech. rep. Stanford University, 2017. URL: `https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2761115.pdf`.

[12] J. Odede and I. Frommholz. "JayBot–Aiding University Students and Admission with an LLM-based Chatbot". In: *Proceedings of the 2024 Conference on Human Information Interaction and Retrieval*. Mar. 2024, pp. 391–395.

[13] Carlos Segura et al. "Chatbol, a chatbot for the Spanish "La Liga"". In: *Proceedings of the 9th International Workshop on Spoken Dialogue System Technology*. Springer, 2019, pp. 319–330.

[14] Iulian V Serban et al. "A Deep Reinforcement Learning Chatbot". In: *arXiv preprint arXiv:1709.02349* (2017).

[15] Ming-Hsiang Su et al. "A chatbot using LSTM-based multi-layer embedding for elderly care". In: *Proceeding of the International Conference on Orange Technologies (ICOT)*. IEEE, 2017, pp. 70–74.

[16] Guido Tascini. *AI-Chatbot Using Deep Learning to Assist the Elderly*. Springer, 2019, pp. 303–315.

[17] Tsung-Hsien Wen et al. "A network-based end-to-end trainable task-oriented dialogue system". In: *arXiv:1604.04562* (2016).

Appendix

	Event	Location	No Entity	Total
Event	10	0	0	10
Location	0	13	0	13
No Entity	0	0	58	58
Total	10	13	58	81

Table 9.1: Entity Confusion Matrix with Totals

	Goodbye	Greet	Search Events	Search Tickets	Total
Goodbye	4	0	0	0	4
Greet	0	4	0	0	4
Search Events	0	0	3	0	3
Search Tickets	0	0	0	5	5
Total	4	4	3	5	16

Table 9.2: Intent Confusion Matrix with Totals

Word counts are more than 11,500

Table 9.3: Word Counts

```
rasa_project/
├── actions/
│   ├── __init__.py
│   └── actions.py              # Custom action code
├── config.yml                  # Configuration file for the NLU and core pipelines
├── data/
│   ├── nlu.yml                 # NLU training data (intents, entities)
│   ├── stories.yml             # Dialogue training stories
│   ├── rules.yml               # Rules for specific actions
│   └── domain.yml              # Domain file (intents, entities, slots, responses, actio
├── models/
│   └── <model_name>.tar.gz     # Trained model files
├── tests/
│   ├── conversation_tests.yml  # Automated tests for conversations
├── credentials.yml             # Credentials for external services (e.g., Slack, Faceboo
├── endpoints.yml               # Endpoints for custom actions and external APIs
├── environment.yml             # Optional: Environment file for dependencies
├── README.md                   # Documentation and project overview
└── Dockerfile                  # Optional: Dockerfile for containerization
```

Figure 9.1: Rasa Project File Structure

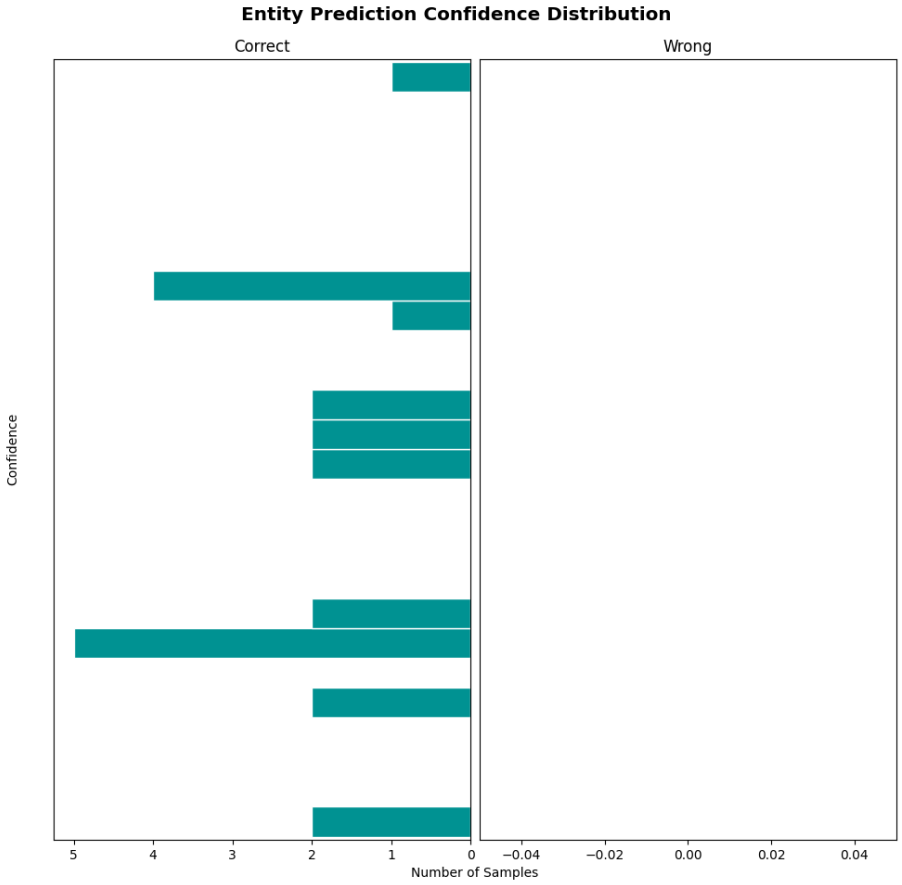

Figure 9.2: Entity Prediction Confidence Distribution

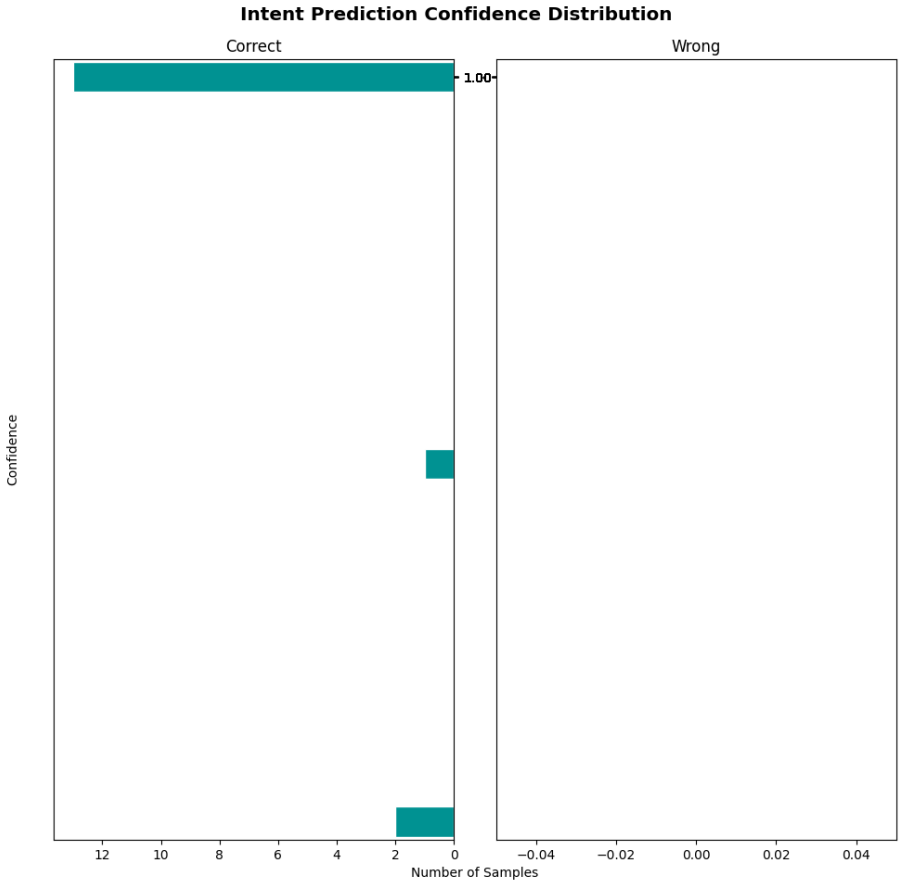

Figure 9.3: Intent Prediction Confidence Distribution