Assignment 4

Chatbot Creation, Implementation, and Improvement

Theodore Fitch

Department of Data Analytics, University of Maryland Global Campus

DATA 650: Big Data Analytics

Dr. Ozan Ozcan

August 6th, Summer 2024

Introduction:

In the rapidly evolving digital landscape, chatbots have emerged as a transformative technology, revolutionizing customer service and interaction. Businesses across various sectors have integrated chatbots to provide instant support, streamline operations, and enhance customer experience. These virtual assistants are available 24/7, offering quick resolutions and freeing up human resources for more complex tasks. With advancements in natural language processing (NLP) and machine learning, chatbots can understand and respond to user inquiries with increasing accuracy and relevance, making them an indispensable tool in the modern business ecosystem (Adamopoulou et al., 2020). The emergence of chatbots has been driven by the need for businesses to keep up with the growing expectations of customers who demand immediate responses and personalized experiences. From handling routine queries to assisting with transactions, chatbots can perform a wide range of functions that improve efficiency and customer satisfaction. The flexibility and scalability of chatbots make them ideal for businesses of all sizes, enabling them to reach and engage with customers more effectively. While there are many concerns about chatbots replacing human jobs, overall chatbots have been proven to be a net positive in returning hours back to human users for using their time on other tasks. The easy litmus test of this is asking how many of us would rather not deal with monotonous emails or messages asking simple questions that could be answered from reading the SOP or manual.

A chatbot was created in order to demonstrate the general workflow of chatbot design, creation, implementation, and improvement. The Vida Rica chatbot was designed specifically for a popular Mexican restaurant chain, aiming to enhance customer interaction by providing a seamless ordering and feedback experience. Vida Rica is a fictional restaurant made only for the

purposes of this demonstration; all menu items, phone numbers, and addresses are made up for the purpose of example only. This chatbot acts as a digital concierge, assisting customers with ordering food, checking store locations and hours, and collecting valuable feedback to improve services. By automating these tasks, Vida Rica ensures that its customers enjoy a consistent and personalized experience, leading to increased satisfaction and loyalty. In particular, customers are greeted by name, given a friendly experience, and even have plenty of relatable jokes, emojis, and videos are part of their experience with the bot so that it feels familiar and warm (Ursu, 2019).

End-users of the Vida Rica chatbot are customers who want to place food orders, inquire about menu options, or provide feedback on their dining experience. Users interact with the chatbot through a conversational interface that mimics human interaction, allowing them to ask questions or place orders in a natural and intuitive manner. The chatbot's ability to handle multiple tasks efficiently provides users with a convenient and engaging experience, which is particularly beneficial and many argue as critical in today's fast-paced digital environment (Dash et al., 2019). For the purpose of this analysis, only a rudimentary design of a chatbot was made – higher order chatbots have the ability to perform incredibly intelligent tasks and interface with multiple databases in order to answer as many questions as possibly in order to free up as many hours from a human agent as possible. Chatbots can have incredibly extensive designs or have only a basic design (having a wide variance of high ceilings and a low basement). This makes them incredibly helpful for all types of businesses to use since mom-and-pop shops are able to afford making/designing a simple chatbot, or large corporations can design intricate chatbots.

It's important to make a critical distinguishing here between chatbots and large language models (LLMs) due to the popularity of both especially in the current cultural zeitgeist. They differ primarily in their scope and application. Chatbots are specialized applications designed to facilitate conversation and perform specific tasks such as customer service, order placement, or information retrieval, typically within a defined domain (Adamopoulou et al., 2020). They are designed and follow a preset, particular dialog flow. In contrast, LLMs, like GPT-4, are advanced AI models trained on vast datasets capable of generating human-like text, understanding complex queries, and providing contextually rich responses across a broad range of topics, making them more versatile but less specialized for specific use cases compared to chatbots (Bratić et al., 2024). Chatbots can leverage LLMs or other AI technologies (like NLP) to provide structured and guided interactions, ensuring users receive relevant information and assistance based on predefined workflows and rules. In contrast, LLMs have a more unstructured approach of general intelligence enabling them to understand and generate nuanced language and perform tasks such as content creation, language translation, and sentiment analysis, offering flexibility in applications beyond just conversational agents (Bratić et al., 2024).

The expected benefits of the Vida Rica chatbot include enhanced customer engagement, improved operational efficiency, and valuable insights into customer preferences and feedback. From an owner’s perspective, it will save each store many hours of answering the phone or answering emails from hungry customers (Dash et al., 2019). By providing instant responses and personalized interactions, the chatbot builds stronger relationships with customers, encouraging repeat visits and fostering brand loyalty (which is becoming increasingly difficult to foster). Additionally, the data collected through the chatbot interactions can be analyzed to identify

trends and areas for improvement, enabling Vida Rica to continually refine its offerings and services.

General Dialog Flow:

The general dialog flow of the Vida Rica chatbot is designed to guide users through various scenarios, ensuring a smooth and efficient interaction. A detailed description of the chatbot design will be discussed later. First, here are five key dialog scenarios:

1. **Ordering Food:** The chatbot assists users in placing food orders by guiding them through a series of questions about their preferences. It prompts for the location, type of food (burrito, nachos, tacos), meat choice (pollo, carne asada, al pastor), and spice level. The chatbot confirms the order details and allows users to make adjustments if needed, ensuring accuracy and customer satisfaction.

2. **Checking Store Hours:** Users can inquire about store hours, and the chatbot provides detailed information about opening and closing times for each location. This helps users plan their visits and ensure that they can place orders or pick up food at their convenience.

3. **Providing Feedback:** The chatbot encourages users to share feedback about their dining experience. It collects email addresses for follow-up and stores the feedback, allowing Vida Rica to address any issues and continually improve its services. This scenario highlights the chatbot's role in maintaining open communication with customers.

4. **Finding Locations:** When users ask about store locations, the chatbot provides addresses and directions for each of Vida Rica's branches. This information helps users find the nearest location and facilitates a smoother ordering or dining experience.

5. **Sharing a Joke:** To enhance user engagement, the chatbot can tell jokes related to Mexican food, adding a touch of humor and personality to the interaction. This feature makes the chatbot experience more enjoyable and memorable for users, encouraging them to return for future interactions.

These dialog scenarios are designed to ensure that users receive prompt, accurate, and engaging responses, making the Vida Rica chatbot an effective tool for enhancing customer interaction and satisfaction. Finally, it's critical to note that many terms will be used throughout this analysis. All of these terms will be used fluidly and won't be redefined in the text as they are defined in Attachment 1 (IBM, 2022). Before proceeding, readers should review the terms to ensure they will understand the continuing text.

Chatbot Design:

The design process for the Vida Rica chatbot involved carefully crafting both user-defined and system entities, as well as comprehensive intents, to ensure that the chatbot can effectively understand and respond to a wide range of user inputs. By utilizing Watson Assistant's capabilities, the chatbot was structured to handle specific tasks such as food ordering and feedback collection while providing an engaging and seamless user experience. During this entire discussion of the design, one should refer to Figure 28 in order to understand any needed vocabulary to understand this analysis. Entities are used to extract specific pieces of information from user input, helping the chatbot understand and respond appropriately. User entities are

custom-defined and specific to the chatbot's domain. They are created to capture essential details that are unique to the interactions the chatbot is designed to handle. In the Vida Rica chatbot, user entities are used to understand specific aspects of a customer's order and preferences.

Entity	Description	Synonyms
@order_location	Captures the location for pickup	"city of angels", "downtown", "dtla", "la" (for Los Angeles), "carlsbad", "encinitas", "oceanside", "sd" (for San Diego), "barbara", "sb", "the coast" (for Santa Barbara)
@order_meat	Captures the type of meat in the order	"carnitas", "pig", "pork" (for al pastor), "asada", "beef", "carne", "cow" (for carne asada), "chicken", "chicky" (for pollo)
@order_type	Captures the type of food being ordered	"burritoes", "burritos" (for burrito), "nachoes", "nachos" (for nacho), "tacos" (for taco)
@final_confirm	Captures the final confirmation from user	"by no means", "nah", "nope" (for no), "affirmative", "certainly", "confirm", "done", "yah", "yea", "yep", "yup" (for yes)
@spicylevel	Captures the user's preferred spiciness level	No synonyms as it uses percentages

Figure 1. All user entities used in the chatbot – entities are designated by the $ symbol.

The design of user entities in the Vida Rica chatbot focuses on capturing key elements of a customer's order to provide a personalized and accurate service. The **@order_location** entity is crucial as it identifies where the customer wishes to pick up their order, accommodating different synonyms for cities like "LA" or "SD" to account for variations in user input. It also accomodates for variation like "Encinitas" or "Carlsbad" since those are suburbs of San Diego that users might incorrectly use instead of San Diego. Similarly, the **@order_meat** entity ensures that the chatbot can accurately recognize and offer the user's preferred protein choice, using synonyms like "asada" for "carne asada" to align with common terminology. It also accomodates this for both Spanish and English since it is common for customers to know and use

both as equivalent. The **@order_type** entity distinguishes between the types of food items available, such as burritos or tacos, and includes plural forms and colloquial terms to match user language. The **@final_confirm** entity supports the chatbot in confirming order details before processing, with synonyms like "yup" for "yes" or "nah" for "no" to reflect casual affirmations or rejections. Finally, the **@spicylevel** entity captures the user's preferred level of spiciness using a numerical scale, ensuring the order meets the customer's taste preferences. These entities, supported by synonyms, ensure flexibility and robustness in understanding and processing user requests, enhancing the chatbot's ability to deliver a seamless ordering experience.

Entity	Description
@sys-date	Captures dates
@sys-number	Captures numerical input
@sys-percentage	Captures percentages
@sys-time	Captures time input

Figure 2. All system entities used in the chatbot – entities are designated by the $ symbol.

System entities in the Vida Rica chatbot are predefined entities provided by Watson Assistant, which facilitate the capture of common data types such as dates, numbers, and percentages without the need for custom configuration. The **@sys-date** entity allows the chatbot to understand and process any date-related inquiries, such as asking when an order will be ready or checking the operating hours on specific days. By using **@sys-number**, the chatbot can interpret numerical inputs, which is essential for scenarios like confirming phone numbers, processing order quantities, or specifying preferred spice levels using the **@spicylevel** entity. This system entity is also instrumental in handling inputs where users might provide numbers in various formats or contexts. The **@sys-percentage** entity is used to gauge users' spice preferences accurately, allowing customers to express their desired heat level using a percentage, thereby personalizing their dining experience. Additionally, **@sys-time** helps in managing time-

related queries, ensuring that customers receive accurate information regarding the best times to place orders or inquire about when their meals will be ready for pickup. It's important to note that **@sys-currency** was also an available option but it was not used in this chatbot, and thus was not added to the table above. By leveraging these system entities, the chatbot can efficiently handle diverse user inputs, providing precise and relevant responses that enhance the overall interaction quality. The utilization of system entities streamlines the development process by reducing the need for extensive custom coding and ensures that the chatbot can handle a broad range of queries effectively, contributing to a seamless and intuitive user experience.

Intents are designed to capture the purpose behind a user's input, allowing the chatbot to determine the appropriate response or action. Each intent should be accompanied by multiple examples to improve the model's ability to recognize varied user expressions.

Intent	Description	Examples
#Email_me	Recognize when a user wants to provide an email	"email", "emailed", "here's my address", "message me here", "please follow up with me"
#Feedback	Recognize when a user is giving feedback	"Angry", "Complaint", "Criticism", "Feedback", "Garbage", "Review", "Sick"
#goodbye	Recognize when a user is ending the interaction	"adios", "bye", "bye-bye", "enjoy your day", "goodbye", "hasta la vista", "have a nice day", "I will be back", "k thanks bye", "later"
#Hours	Recognize inquiries about store hours	"Are you open right now", "Date", "Day", "Days", "Friday", "Hours open", "Hours to order", "Monday", "Saturday", "Sunday", "Thursday", "Tuesday", "Wednesday", "Week", "What are your hours", "What hours are you open", "What time of day can I get food", "When can I order food"
#joke	Recognize when a user asks for a joke	"another joke", "funny", "hahahaha", "joke", "pun"
#Locations	Recognize inquiries about store locations	"What is the store address?", "What's the address?", "Where are your locations?", "Where are your stores?", "Where is pick up?"
#menu	Recognize when a user asks for the menu	"Can I see your menu?", "Show me the food you have", "Show me the menu", "What food do you have?", "What's on the menu?", "What type of food can I order?"
#Order	Recognize when a user wants to place an order	"Hungry", "I'm hungry and want food", "I want a bite to eat", "I want to order", "Order food", "Order food online", "Order online", "Order pick up", "Order pickup", "Want food"

Figure 3. All intents used in the chatbot – intents are designated by the # symbol.

The design of intents in the Vida Rica chatbot is crucial to understanding user intentions and ensuring smooth interaction with the system. Each intent is carefully crafted to address specific scenarios that customers might encounter while interacting with the chatbot. For instance, the **#Email_me** intent identifies when a user wants to provide or receive information via email, which is vital for follow-ups and personalized communication. The **#Feedback** intent is designed to capture any user input related to their dining experience, enabling Vida Rica to gather valuable insights and improve their service. The **#goodbye** intent helps in gracefully concluding interactions when a user indicates they are done with their inquiries. The **#Hours** and **#Locations** intents address common customer queries about the restaurant's operational hours and physical addresses, ensuring that users can access essential information easily.

The **#joke** intent adds a layer of engagement and personality to the interaction, allowing users to enjoy a lighthearted moment and enhancing user satisfaction (Ursu, 2019). Again, this helps foster brand loyalty, generational marketing, and reputation. The **#menu** intent provides a pathway for users to explore the available food options, guiding them through the ordering process or to know what the restaurant has before visiting. Finally, the **#Order** intent is central to the chatbot's functionality, as it initiates the order placement process and ensures that users can specify their preferences, such as type of food, location, and other order details. Each intent is linked to specific dialog nodes within the flow, allowing the chatbot to trigger appropriate responses and actions based on the detected intent. While the general dialog is structured such that there should be a general flow for users to make it easy to navigate where they need to go, if a user were to type in something unexpected, these intents should be able to guide the customer to the right node to address their need. This structured approach ensures that the chatbot can handle various inquiries seamlessly, providing accurate responses and enhancing the overall customer experience. By defining intents with multiple examples, the chatbot can accurately recognize diverse expressions of the same intention, improving its ability to respond effectively

to user needs.

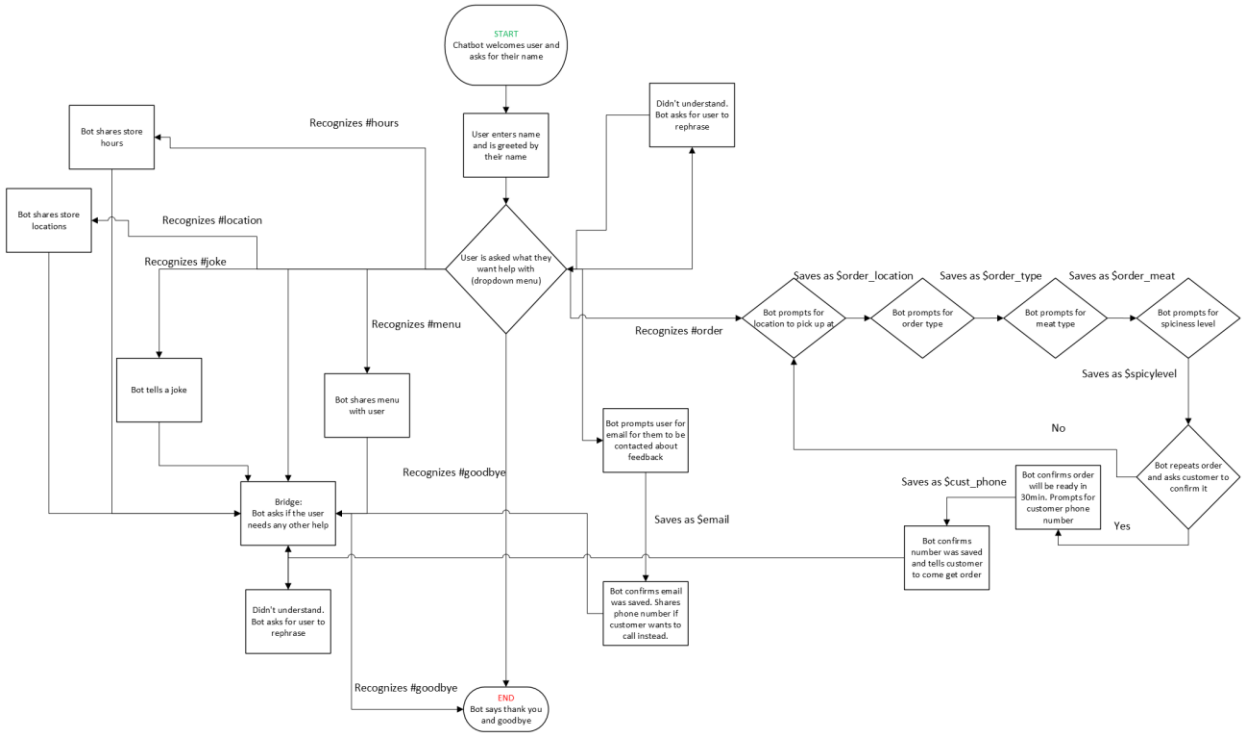

Figure 4. The overall dialog flow of the Vida Rica chatbot shows how the entities and intents interact together to guide the users to the information they need or the action they need to take (generated via Visio).

The dialog flow of the Vida Rica chatbot is designed to guide users through a variety of interactions, ensuring they receive the information and assistance they need efficiently. The flow begins with the chatbot welcoming the user and asking for their name, which is stored as a context variable **$name**. This initial interaction sets a personalized tone, as the chatbot will use the user's name throughout the conversation to create a friendly and engaging experience.

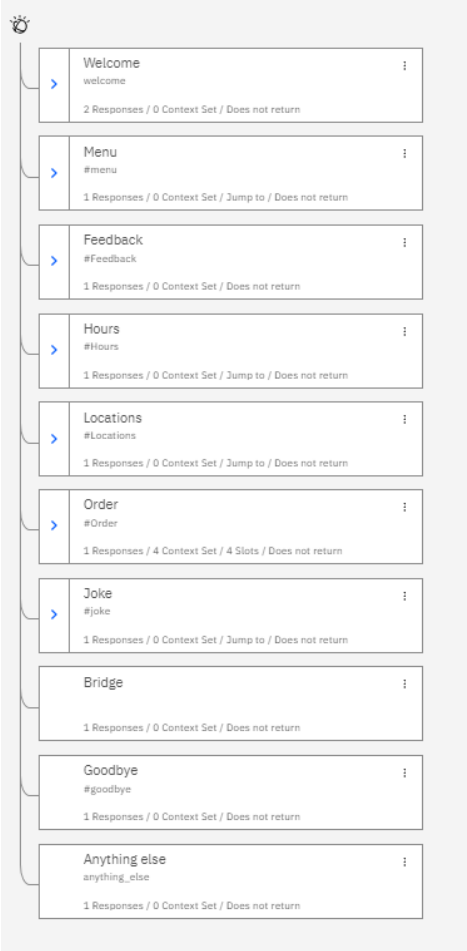

Figure 5. Vida Rica dialog flow shown via the Watson nodes. This is how the chatbot is designed.

Once the user has provided their name, the chatbot asks what they would like help with, presenting options through a dropdown menu. This menu includes intents such as **#Hours**, **#Locations**, **#joke**, **#menu**, **#Order**, and **#Feedback**. Each intent is linked to a specific dialog path, allowing the chatbot to tailor its responses based on the user's choice. For instance, if the user selects **#Hours**, the chatbot provides the operating hours for all store locations, leveraging the intent to offer relevant information. Similarly, selecting **#Locations** prompts the chatbot to share addresses and directions, fulfilling the user's inquiry. It should be noted that from this welcoming node, users can either interact with the dropdown menu or they can type their question or keywords into the chat and either one will help move the chatbot forward. If the bot

doesn't recognize the input, then it will prompt the users to rephrase their request (and can offer possible intents for the user to choose from).

In the case of an order, the chatbot follows a structured sequence to collect necessary details, storing them as context variables for a seamless transaction. The process involves recognizing the **#Order** intent and subsequently capturing details such as **$order_location**, **$order_type**, **$order_meat**, and **$spicylevel**. The chatbot prompts the user for each piece of information, ensuring clarity and accuracy. These variables are essential for confirming the order and providing a summary to the user, enhancing the accuracy of the service provided. As previously noted, these prompts to capture details have plenty of variability and flexibility built into the entities. For example, a customer could use SD, sd, San Diego, or Encinitas in order to trigger the bot to select the San Diego location. This is true of each of these entities. Critically, the meat types are programmed to be accepted regardless of if the user inputs the English or Spanish name for any of the meat types (since common parlance in California uses them interchangeably). If the order is not confirmed, then the bot cancels the order, and resets all of the previously stored variables back to null. It then directs customers to let it know if they'd like to make another/different order. If the order is confirmed, the chatbot asks for a contact number, stored as **$cust_phone**, to ensure communication in case of any issues with the order. It's critical to note here that this variable stores whatever the customer inputs; it does not check to see if this is a valid phone number (like xxx-xxx-xxxx). It is possible to program chatbots like this to only accept valid inputs following particular patterns. This is described as a pattern entity in which the chatbot doesn't look for similar words or synonyms, but instead searches for a pattern to recognize an entity. This was attempted multiple times using a classroom provided example in order to recognize an email using the pattern entity.

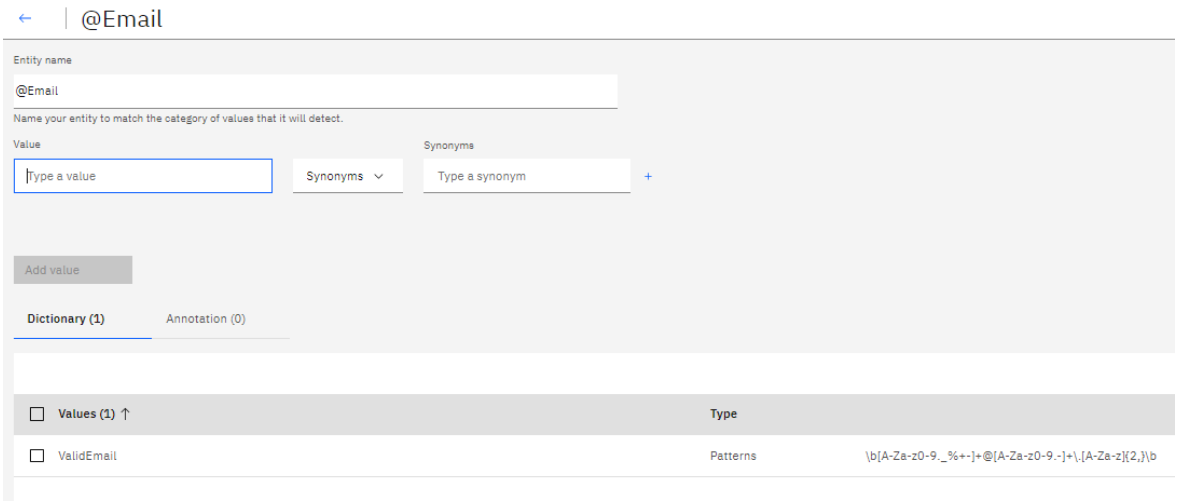

Figure 6. The Watson interface showing the entity for @email being a pattern entity searching for a particular pattern.

Unfortunately, even with requesting TA help, this entity did not work correctly and it is unknown why. Therefore, no pattern entities were further sought after in order to recognize inputs like a valid phone number. Ideally, a chatbot should use a pattern entity in order to validate correct entries are made for examples like these (phone number, name, email address, mailing address, etc.). The dialog flow also incorporates error handling and alternative paths to address unexpected inputs. If the chatbot doesn't understand a user's response, it asks the user to rephrase, ensuring the conversation stays on track. Additionally, the **#Feedback** intent enables users to share their thoughts or concerns, with the chatbot prompting for an email address to facilitate follow-up communication, storing this input as **$email**. Again, this relies on the user making a valid input but as a failsafe a phone number to call customer service is also added after storing the users email. This feedback loop is vital for Vida Rica to improve its services based on customer insights. It also allows them to retain customers' emails in order to offer promos, advertisements, or reparations if they had a bad experience.

The flow includes a **#goodbye** intent, which allows the chatbot to conclude the interaction gracefully. Upon recognizing this intent, the chatbot thanks the user for their visit and

says goodbye, providing a satisfying end to the conversation. This well-structured dialog flow ensures that users can navigate the chatbot with ease, accessing the information they need and completing transactions efficiently, while also providing a personal touch to enhance the overall customer experience.

Finally, the bridge node in the Vida Rica chatbot serves as a pivotal element that enhances the user experience by seamlessly connecting different parts of the conversation and ensuring that all user needs are addressed before the session ends. Positioned after key interactions, such as providing information or completing an order, the bridge node checks in with the user to see if there is anything else they require assistance with. This node effectively acts as a transition point, guiding users back to the main menu or allowing them to explore additional options, such as reviewing the menu or providing feedback. Most of the nodes onces chosen will redirect automatically back to this node so that the bot moves the conversation forward. By asking, "Is there anything else I can help with today?" the bridge node not only maintains the flow of the conversation but also encourages users to fully utilize the chatbot's capabilities. The bridge node exemplifies the chatbot’s commitment to delivering thorough service, inviting users to explore more of what Vida Rica offers while ensuring that all their questions are answered.

While this concludes the discussion of the general flow of the dialog, several other key design features need to be discussed. For example, the system was set up to recognize the time from the outset, and then it conditionally welcomes the users based on the time of day. This type of context makes the bot feel more human, aware, and welcoming. While conditional responses could be applied to any node, it was only applied to this one as the welcome experience for users is one of the most important in order to begin engagement with customers. Conditional responses

aren't necessary for many other nodes like **#menu** or **#location**. One area they would commonly be applied is based upon the customer's order to show a photo of what order they've chosen (in this case a burrito, nachos, or tacos).

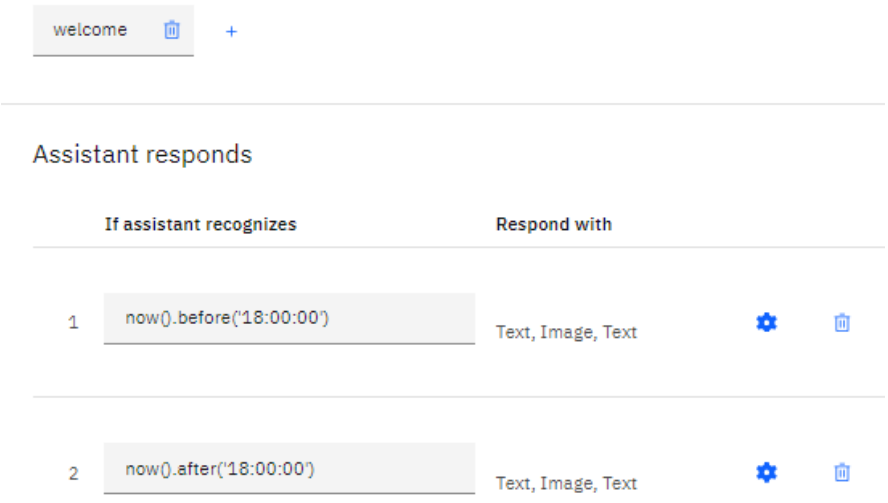

Figure 7. The Watson interface showing the conditional welcome response.

Furthermore, emojis were used throughout the bot in order to both make the bot feel more human and warmer but more importantly it makes the interface easier to use. It makes it easier for whom English or Spanish is not someone's first language, and it also is an easy visual cue to prompt humans to remember more easily. Additionally, images were used throughout the entire bot from the welcome node, order node, menu, and goodbye node. This is important for a restaurant for customers to visualize the kind of food that they will receive (hunger is a good motivator to buy more food) but also to keep users engaged with the chat. It will be discussed later but the biggest KPIs of a chatbot like this are user engagement, questions answered, and hours returned back to the business.

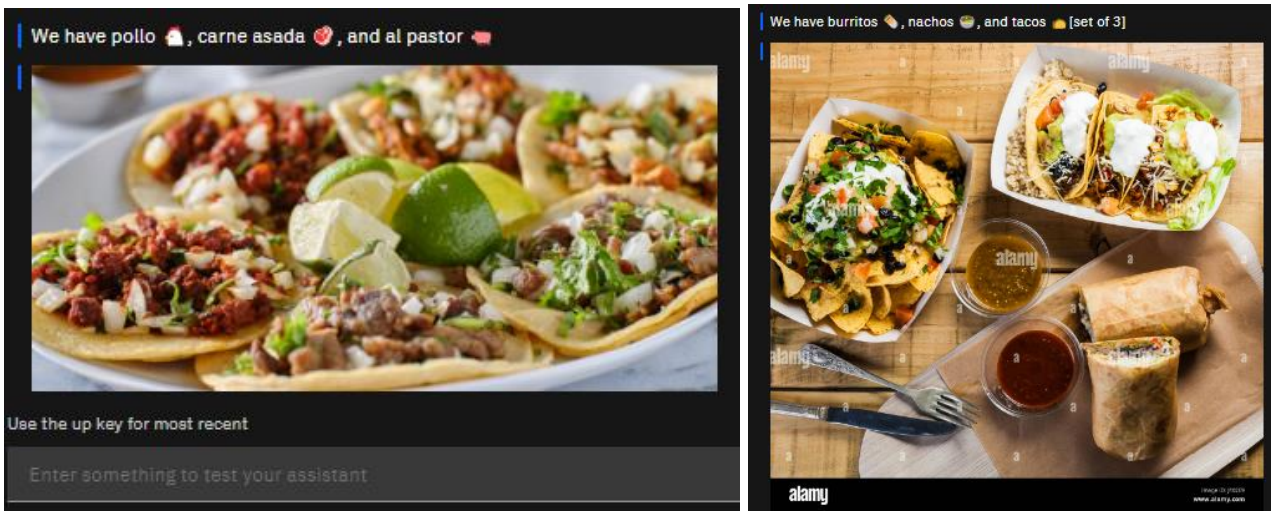

Figure 8. The Watson chatbot testing interface showing the images and emojis being used.

Another key feature to highlight is the ability for Watson to allow for varied responses in the same node. This allows the chatbot to feel like it is more human and offers more variability to the customer. Having a novel experience each time they use the bot fosters a sense of personality as though they're not just talking to a computer. It addition, a video was added directly into the interface of the chat – in this case, a video containing mariachi music was added so that the final experience the user has will be a fun one further elevating the brand for customers. Having a video also helps with engagement where customers may want to stay with the chatbot longer and ask more questions, or ideally, make another order.

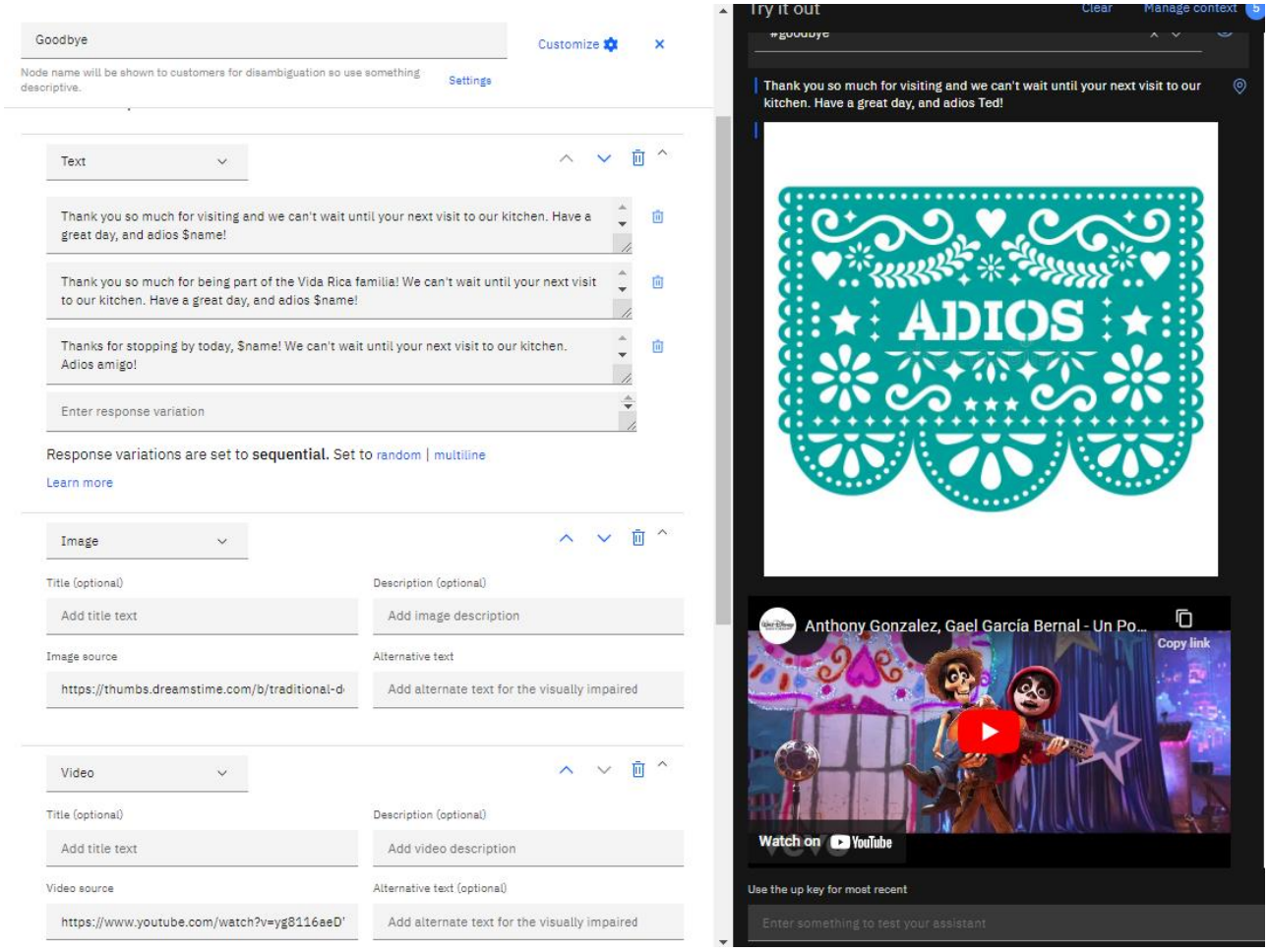

Figure 9. The Watson chatbot testing interface showing video, images, and varied responses being used.

Furthermore, slots were used in order to collect context variables from customers in the order workflow. Each prompt searches for a previously discussed entity, when it hasn't found it, the bot prompts the user to enter each respective variable it needs to complete the order. Ideally, these would be sent to an established database where orders can be fulfilled for each location. In addition, it would be ideal for each order to come with an order number such that users can inquire about status or get updates (which is a common feature on many modern apps like Grubhub or Doordash).

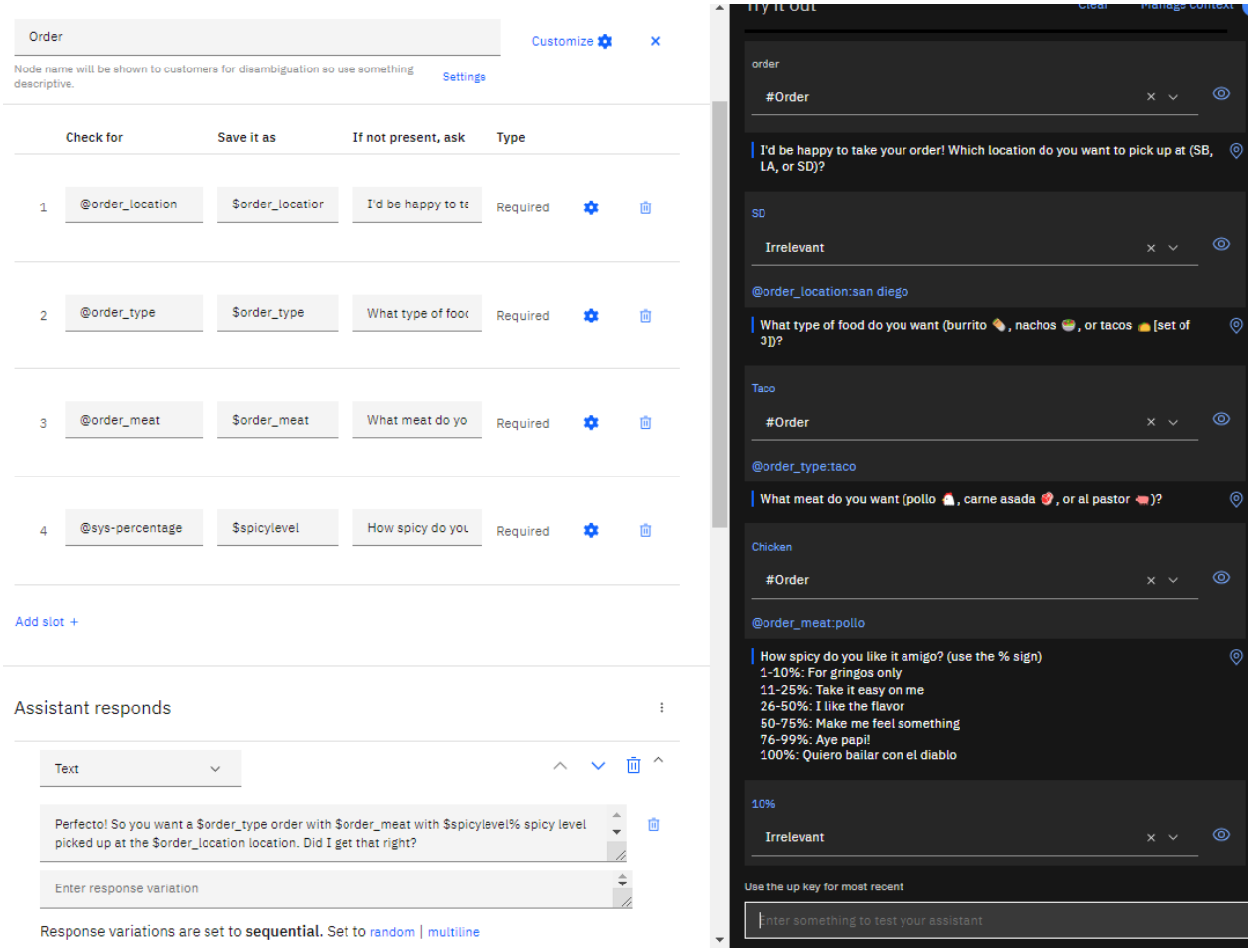

Figure 10. The Watson chatbot testing interface showing multiples slots in order to save the customer's input as a context variable.

It also should be highlighted how the system is able to offer users an option instead of asking them for free text inputs. In the below scenario, users can either select a button for Yes or No in order to confirm their order. As previously discussed, they can also type in their response and the system will recognize it (even if it is a "yeah", "ya", "yes", "yesh", etc.) based on the defined user entities. This was also set up for the original welcome menu where each major intent was also set up as an option to make it easier for users to navigate the chatbot. That way users have their options immediately narrowed down to what is most relevant, and importantly it shows the options so that when prompted they can ask their relevant questions. While more

complex chatbots might have an LLM integrated such that it can recognize and address anything a customer would input, chatbots for smaller businesses may only require having a few simpler inputs which makes it more cost effective for the business and easier for the customer.

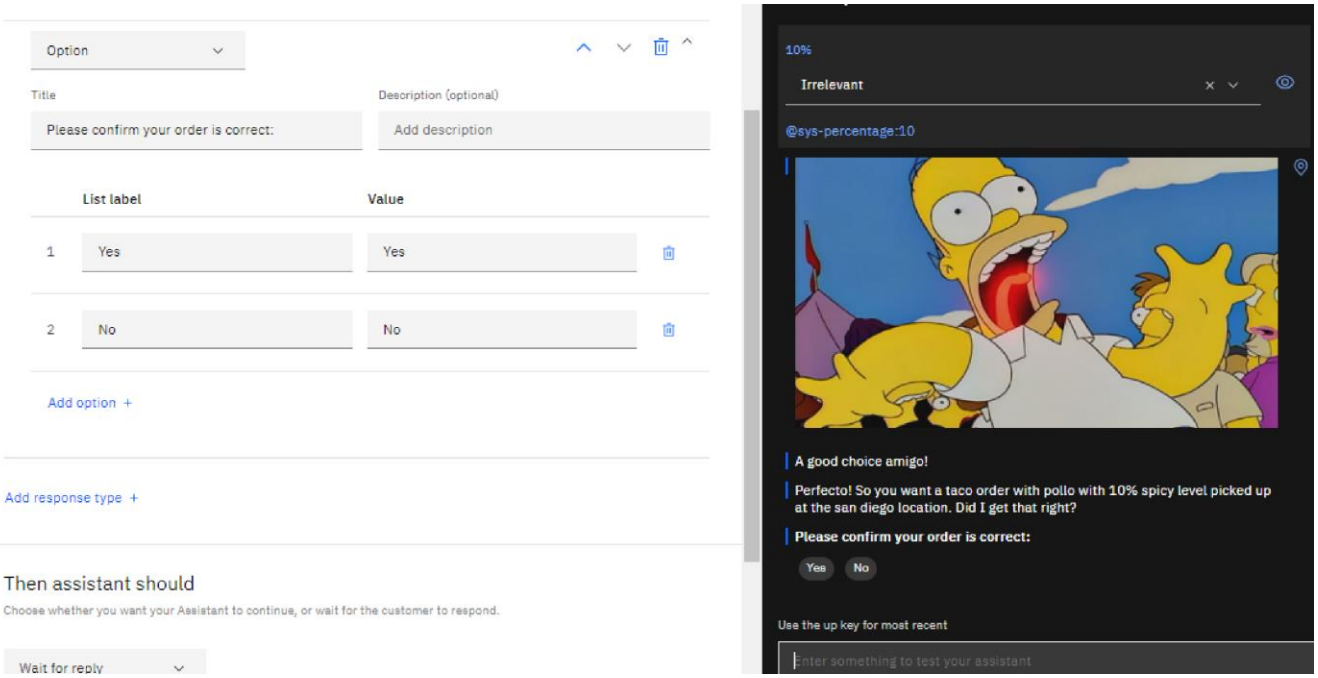

Figure 11. The Watson chatbot testing interface showing an option response type for users to take advantage of.

In the Vida Rica chatbot, handlers and regular expressions were not explicitly utilized, primarily due to the design's focus on leveraging Watson Assistant's core capabilities, such as intents, entities, and context management (however they will be used later as an improvement feature). These features effectively manage user inputs without adding unnecessary complexity. The chatbot's interactions are structured around predefined options and guided conversations, such as ordering food or providing feedback. This structured approach allows the chatbot to use menu options and context variables efficiently, reducing the need for handlers, which are typically used for managing more dynamic or unstructured input scenarios. The use of synonyms within user entities plays a crucial role in capturing variations in user input, making it unnecessary to use regular expressions for parsing different phrases or expressions. This was also

discussed earlier in how emails and phone numbers could have been recognized via pattern entities but instead were chosen not to.

Using the "Try it Out" panel in Watson, I tested various dialog scenarios to ensure the chatbot responded accurately and intuitively to user inputs. From having only a welcome and goodbye node, to having 10 nodes prior to testing. Each node was tested as it was developed in order to ensure the chatbot was acting as it was expected to and errors could be immediately corrected based on their respective node. For example, the chatbot was able to easily store the user's name as a context variable with no issue whatsoever. However, just as it was previously discussed, it was extremely difficult to use the pattern entity in order to have the user input and save their email. Since the pattern entity didn't work, it was instead opted to use a "user input" stored as context variable. The downside to this is that the chatbot will not recognize when something other than a valid email has been input.

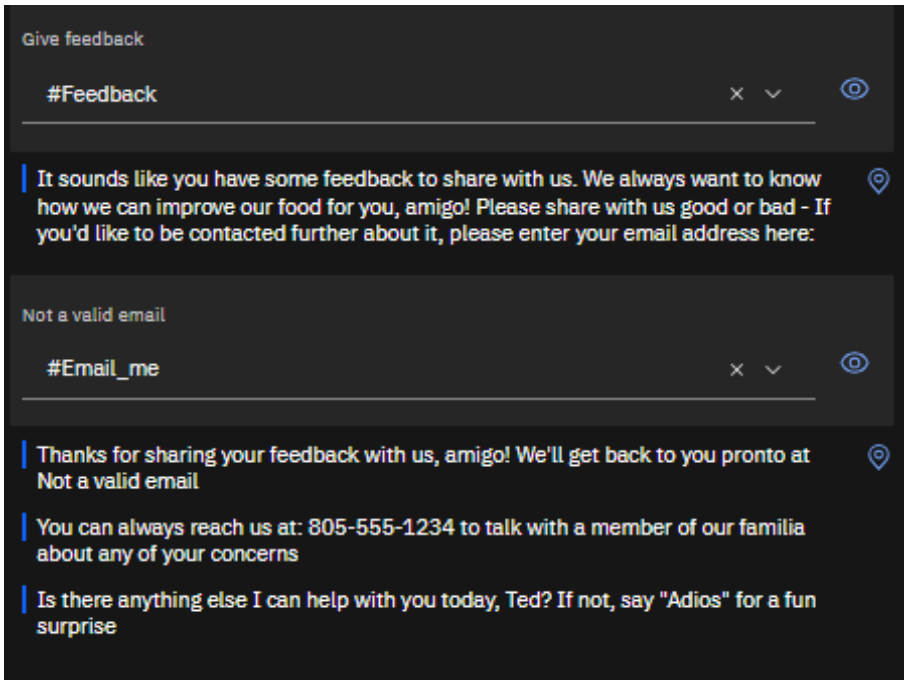

Figure 12. The Watson chatbot testing interface showing accepting a non-email variable saved as the email input.

In addition, another area of pain in development was determining how the spiciness level should be saved whether as a system entity of percentage or number. Because users were thought to be more likely to enter using the percentage sign after the number since they were prompted using percentage ranges, it was decided to use this. Thus, entering a number without the % would not move the bot forward. In an ideal situation, a bot should allow this to be a multiple-choice button to make entry easier for customers. But this percentage choice was chosen in order to demonstrate how the **@sys-percentage** is typically used by the system.

Figure 13. The Watson chatbot testing interface showing how the bot would not accept a number but would accept a percentage input.

Another pain point in development was determining the correct flow of order correction. Originally, it was thought that when a customer told the bot the order was incorrect, the bot should then ask which variable was incorrect to let the customer reinput that variable only. That would be a valid approach for a more complicated bot which allowed for multiple orders at once and with more variation. But for the sake of flow, and because this was a relatively short and simple order system, it was instead decided to take a simpler route. When an incorrect order is confirmed, the bot then erases the 4 context variables redirects the users back to the start of the order workflow.

Another unique design feature about this workflow is how certain inputs are forced in this chatbot. Several workflows will force a valid entry before allowing the users to move on to any other node. These are collecting the user's phone number and email and collecting the order (each variable will have to be collected before the bot will allow a jump to bridge). This was designed as a business decision such that data could be collected. It would be ideal to collect as many customer emails and phone numbers as possible. Likewise, knowing what are the most common order types, meats, spiciness levels, and locations is incredibly important business information. This data could be used in a variety of ways including to know what food is most popular at each location, what location is most popular, and what food types people are most interested in in in general.

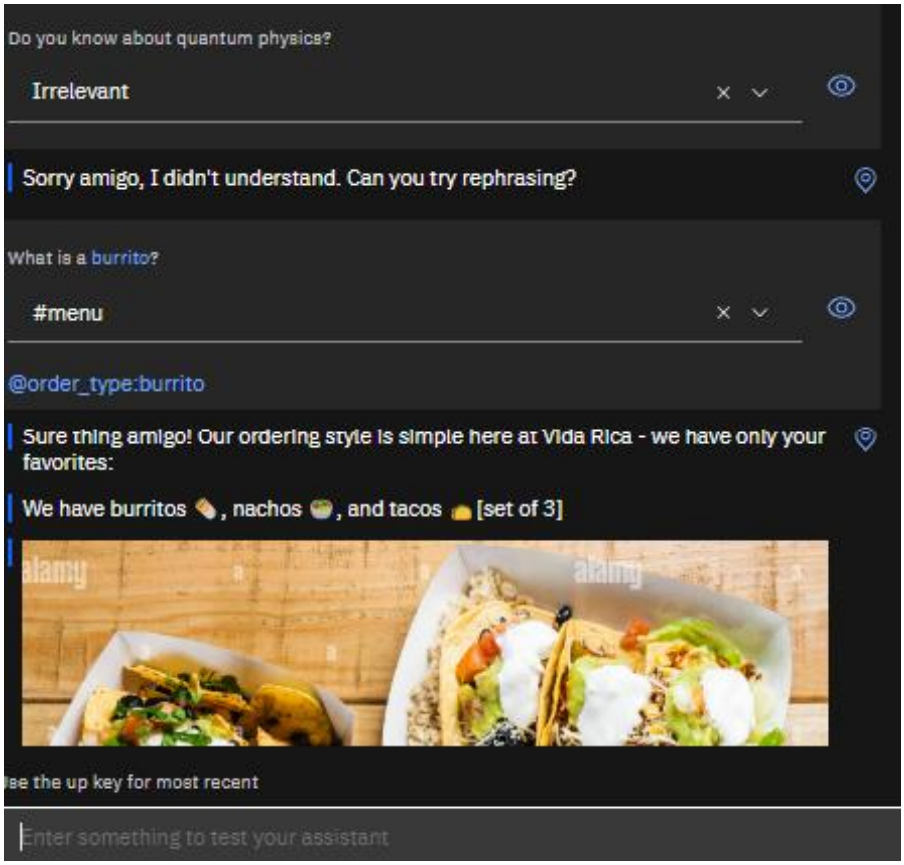

Figure 14. The Watson chatbot testing interface showing how the bot would handle a question it doesn't recognize and a question it does recognize.

Numerous attempts were made to "break" this chatbot by using keywords, hard to handle questions, and irregular inputs. In the example above, the bot doesn't recognize the question regarding quantum physics and thus it jumps to the **anything_else** node causing it to ask the user to rephrase their input. This is commonly needed when designing chatbots in order to accommodate for when the bot doesn't recognize the user's input. For more complicated chatbots, this node should also have programming to recognize the number of attempts made where an incorrect input was made such that on the 2nd or 3rd entry, the bot redirects to user (to an FAQ, a human agent, or another page). Likewise, the question "what is a burrito?" is easily handled by the bot by moving the user to the menu node. NLP and predicting how users will likely respond to a chatbot is a whole field in itself. Users may respond with whole sentences,

single words, or only keywords. We have so much variation in how we communicate and thus it's important to create handling for that in chatbots to account for that variance.

Overall, the development and testing process ensured that the Vida Rica chatbot was equipped to handle user interactions effectively, providing a seamless and engaging experience. By leveraging Watson Assistant's capabilities, we created a chatbot that not only met customer needs but also enhanced the overall service offering of Vida Rica. However, this was only the first part of development where only I tested it as the designer. Next, it needed to be tested by others in order to improve the design and dialog flow. It's critical to understand how it can be broken in order to create guardrails and answer user's questions before they're even asked. This establishes further brand loyalty and customer trust.

Conversation Data Analysis:

Day 1:

In order to pressure test and improve the chatbot, several users tested it out over 2 days to gain insight into how it was working, implement some improvements, and then see how the improvements worked. The chatbot was made into a live link (hyperlinked here) which was provided to testers along with thorough instructions on how to test the bot. The instructions for users can be found in Attachment 2 in this assignment. The experiment was run over Aug 3rd 2024 and Aug 4th. In total from day 1, 11 conversations were had with 5 users where average messages were 13.36. It had a 98.6% coverage with only 2 messages the bot was not confident that it could handle. The top intents by far were **#goodbye**, **#order**, and **#menu** in that order. Likewise, the top entities were **@final_confirm** and **@sys-number** though all of the others were very close behind unlike the intents.

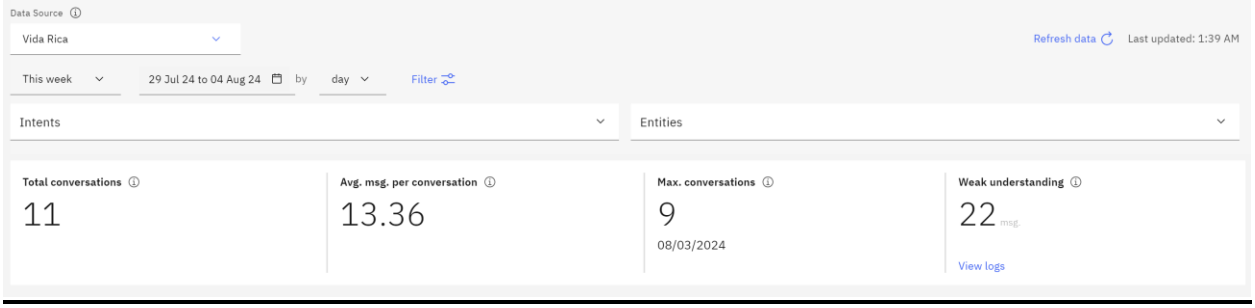

Figure 15. The Watson chatbot testing analytics overview from Day 1.

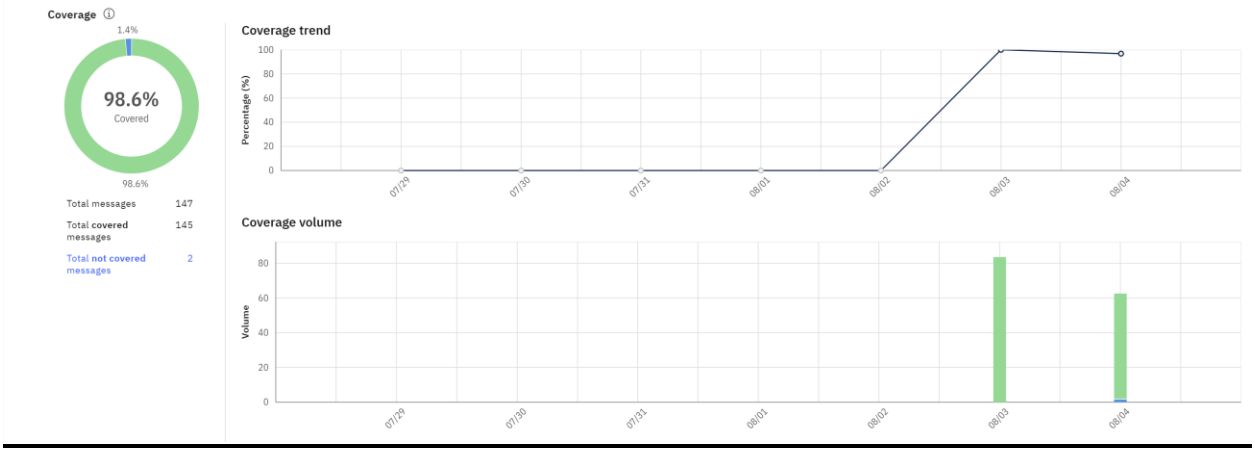

Figure 16. The Watson chatbot testing analytics of coverage from Day 1.

Figure 17. The Watson chatbot testing analytics of conversations and messages from Day 1.

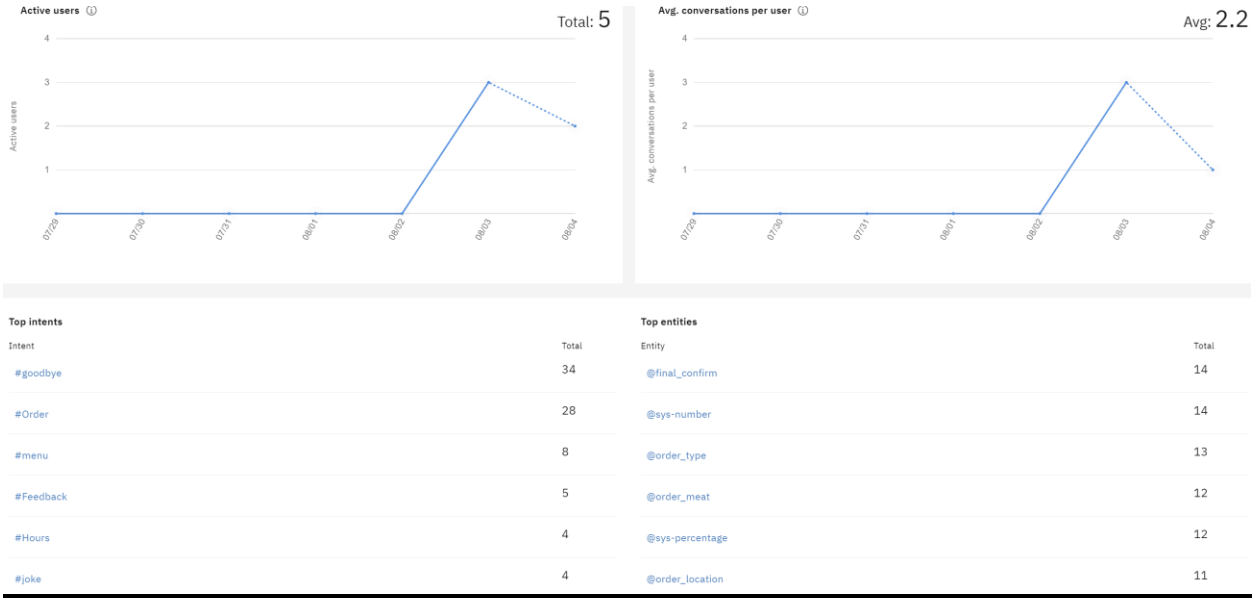

Figure 18. The Watson chatbot testing analytics of users, conversations per user, intents, and entities from Day 1.

In reviewing the chatlogs, the conversations made by the testers highlighted some critical areas of the bot which needed improving. First, validation was required in order to move forward. Several entities/variables needed validation since the testers were able to put invalid emails, phone numbers, or spice levels into the bot and they were accepted. Second, the bot appeared to struggle with several common types of inquiries mostly pertaining to other types of menu items or getting stuck with a particular workflow and not being able to return to the bridge node. Several misclassified intents were updated within the logs to retrain the bot for the next day of testing. A new version of the Vida Rica chatbot was made based upon the review of chatlogs and the input of the testers. The latest version of the chatbot introduces several enhancements aimed at providing a more robust and user-friendly experience. Three users provided in-depth written feedback which were incorporated into this new bot version. All of the testers had a strong IT and software background so they were used to performing QA testing on similar systems such as this. They confirmed the tone of voice felt friendly and established a

warmth to the bot. But they also reported some technical issues which required being addressed, as well as some helpful inputs.

First, in order to help users access the bridge node more easily, a new intent was made for **#bridge** which included synonyms like “unsure” and “help” such that if customers were confused or stuck, they would be redirected to that node. That way if a user become frustrated during the **anything_else** node and they entered something like “help”, they would be redirected. Another recommendation was to add a pause after the goodbye node was reached, then add a pause, then have the bot give customers a promo code if they stuck around longer. This rewards customers for engaging with the bot and digging deeper into conversation with it. It also further fosters brand loyalty. All of their other feedback pertained to how the bot handled errors.

A new **#FAQ** intent was generated and captures inquiries related to dietary restrictions and menu items, providing users with quick access to essential information without needing to navigate through multiple interactions. This was one area where several messages which weren’t covered were added to this intent in order to retrain the bot. There were many interactions observed in which users asked about dietary restrictions or tried to order items not on the menu. They also commonly asked about drinks. Thus, this intent reduces the need for human intervention, streamlining operations and allowing the business to handle more inquiries simultaneously, thereby enhancing efficiency and customer satisfaction.

Additionally, the enhanced use of slots and event handlers facilitates more effective management of conversation flow and input validation. For instance, the chatbot now includes logic to reset the **$spicylevel** variable if a percentage outside the 0-100% range is entered, ensuring users provide valid input before proceeding. Several users tried inputting levels outside the allowed range (from -41% to 1,000% both of which are spicy levels which wouldn’t be easily

interpretable for an order). This prevents errors in order processing and ensures that customer preferences are accurately captured, enhancing the personalization and satisfaction of the service. This also demonstrates how a handler was implemented in order to give multiple conditions for how the bot should proceed based on the user's input.

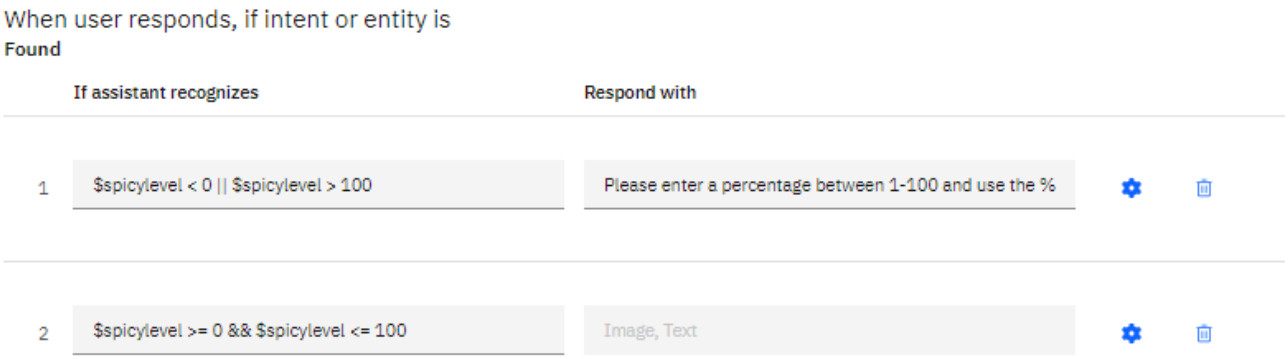

Figure 19. The Watson chatbot designed showing a handler for spiciness level ensuring the variable entered by the user is within a valid range before proceeding.

Similarly, the most important improvements to the chatbot's design were changing the email and phone number entries to ensure they are valid before allowing the user to proceed. The chatbot now uses regular expressions to validate the email addresses and phone numbers, captured through the **@Email** and **@phonenum** entities. The example provided by the classroom as previously discussed would not work. However, a different entity pattern was found made by Mizuguchi (2017). By implementing the patterns **\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b** for emails and **(\d{3})-(\d{3})-(\d{4})** for phone numbers, the chatbot ensures that only valid data is collected, reducing errors and enhancing data integrity. This is crucial for tasks like sending order confirmations or following up on feedback, where accuracy is paramount. The use of regular expressions ensures that users are prompted to correct their inputs if they deviate from the expected format, which helps maintain the quality of interactions and the reliability of the service provided.

Day 2:

The results of the experiment were quite interesting from Day 2 highlighting how the fixes implemented were largely helpful and how there needs to be more improvement. First, based on the overview analytics, there were 13 conversations with a higher average message number of 16.08. There were also only 19 weak understanding messages whereas day 1 had 22 messages. Coverage dropped from 98.6% to 96.2% with a total of 8 messages not covered. But at 209 total messages coverage still remains relatively high. Lastly, the number of users jumped to 9 with 1.44 conversations per user. But this is likely due to usage of a VPN or opening and reopening a browser since the link for testing was given to less than 9 people. It also explains why the number of conversations per user dropped by half as well.

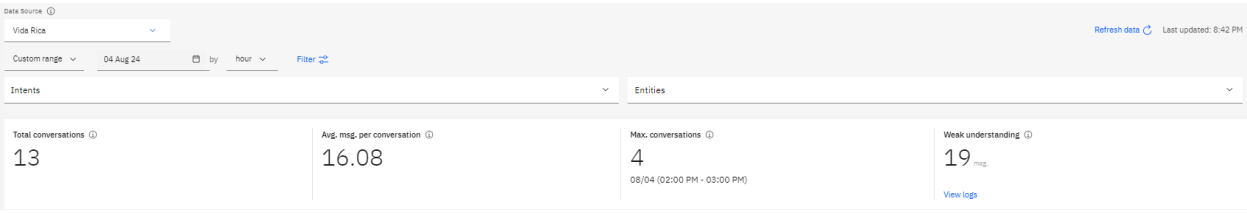

Figure 20. The Watson chatbot testing analytics overview from Day 2.

Figure 21. The Watson chatbot testing analytics of coverage from Day 2.

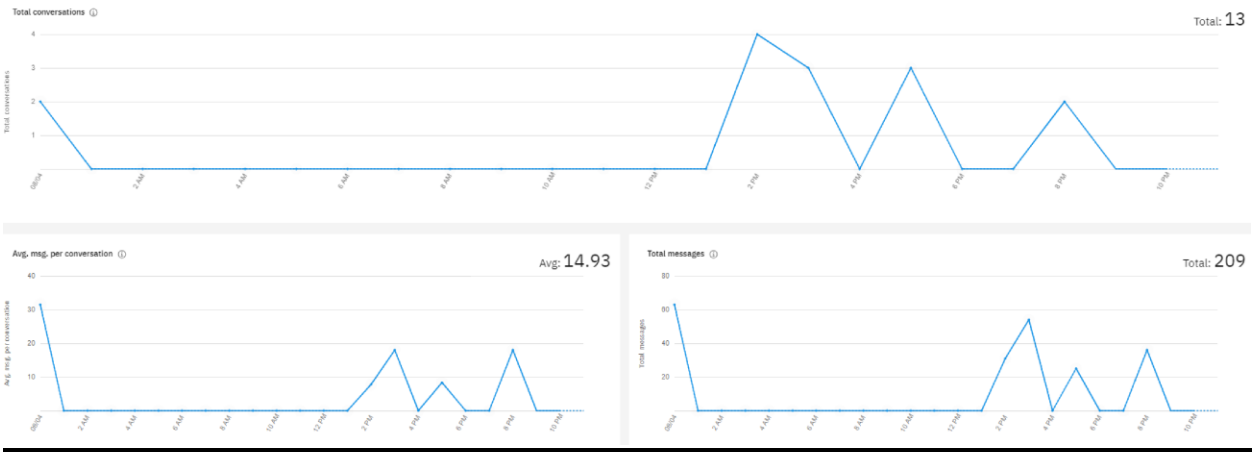

Figure 22. The Watson chatbot testing analytics of conversations and messages from Day 2.

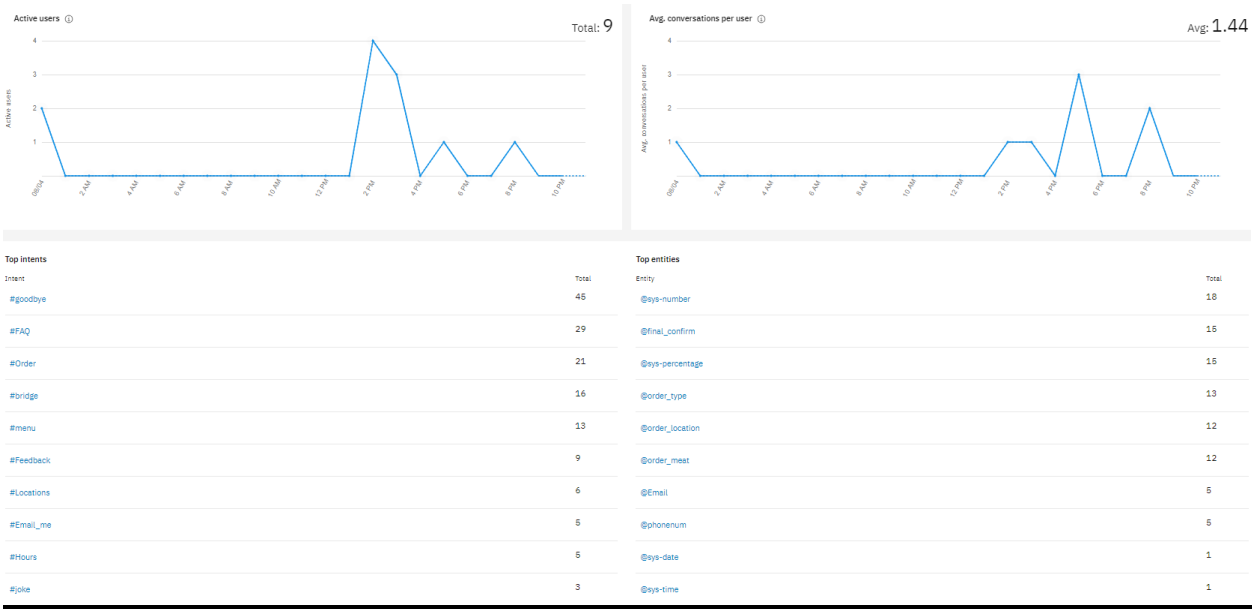

Figure 23. The Watson chatbot testing analytics of users, conversations per user, intents, and entities from Day 2.

The chatlogs show that the changes implemented from day 1 clearly worked overall. The second most common intent was now **#FAQ** at 29 surpassed only by **#goodbye** at 48. **#Order**, **#bridge**, and **#menu** remain near the top as well. The analytics above clearly show that the chatbot had a similar distribution of conversations and users across the days where the bot was tested evenly. In checking the chatlogs, the bot overall performed very well. There were a few

areas where the bot needed some retraining. First, it attempted to classify "about" as **#feedback** and **#bridge** incorrectly. This was adjusted and "about" was added to the **#FAQ** intent.

Figure 24. The Watson chatbot logs from Day 2 showing the bot misclassify "about" as **#feedback** and **#bridge**.

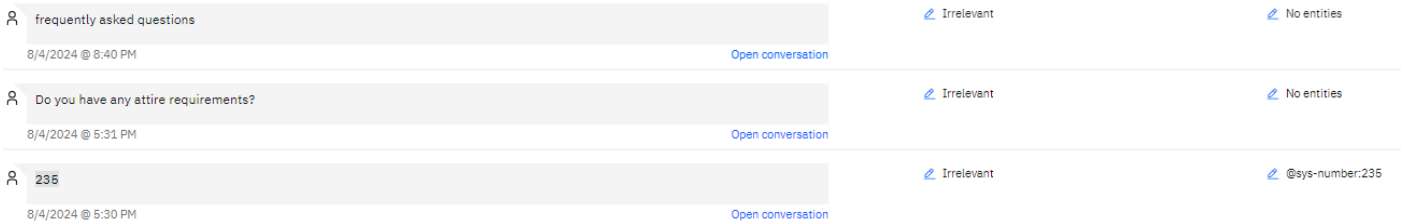

Figure 25. The Watson chatbot logs from Day 2 showing the bot attempting to handle 3 different inputs.

In addition, it was found that "frequently asked questions" was not added to the **#FAQ** intent so this was also updated. Next, the chatbot wasn't able to classify a question pertaining to attire requirements for the restaurant. This would be a rarer question, but something to consider updating within **#FAQ.** Finally, a user also input a seemingly random number (it appears to be an attempt to test the chatbot randomly). The bot correctly recognized the system entity and correctly did not store it or recognize an intent as there is no programming to do so.

The distribution of intents and entities represent fairly well a typical day for a chatbot like this. The vast majority of customers are looking for information or are looking to order. With the current digital landscape, most logistic information like store hours, location, and menu are easily accessible using Google Maps, Yelp, or are typically stored on a static webpage and easily found. Where this chatbot could really help the business is in providing a personalized and interactive customer service experience that goes beyond what static information sources can offer. By integrating the chatbot into the customer journey, businesses can offer immediate

assistance and engage users in a conversational manner, answering queries and providing recommendations tailored to individual preferences. For instance, the chatbot can facilitate the ordering process by guiding customers through menu options, suggesting popular items, and customizing orders based on past interactions, thereby enhancing the overall dining experience. Moreover, the chatbot can serve as a powerful, automated tool for collecting valuable customer feedback and insights. By engaging users in conversation and prompting them for feedback at various touchpoints, the chatbot can gather real-time data on customer preferences and satisfaction levels. It could also ask them personalized questions based upon their order, their questions, or length of patronage. This information can be leveraged to make informed business decisions, such as adjusting menu offerings or enhancing service quality. Additionally, the chatbot can offer promotional opportunities, such as personalized discounts or loyalty rewards, based on user interaction patterns. The information from feedback could be engineered to automatically move the data from the chatbot into a database for later review by the business. This approach not only fosters customer loyalty but also encourages repeat business by creating a dynamic and engaging customer relationship that static information alone cannot achieve. By harnessing the capabilities of AI-driven chatbots, businesses can transform the way they interact with customers, providing a seamless and enriched experience that sets them apart from competitors.

The two-day analysis of the Vida Rica chatbot was an exploration of its functionalities, aimed at understanding its strengths and improving it. Multiple users engaged in various conversations with the chatbot, covering all aspects of the chatbot, from ordering food and providing feedback. Day 1 provided perfect feedback on how to improve this chatbot like its handling of email and phone number inputs, spiciness level validation, its ability to bridge

different conversation paths, and the addition of the **#FAQ** node. These insights prompted significant enhancements which were then seen in the users' chatlogs from Day 2 in which they validated the improvements worked and highlighted some areas for further improvement.

If the experiment were to extend to Day 3, further updates could focus on enhancing the chatbot's conversational depth and engagement. First, would be updating the feedback process to allow users to choose between entering their email to be contacted or directly providing feedback to the chatbot, with this feedback being stored as a text input variable. This would streamline the feedback process and ensure customer insights are captured in an automated fashion. Additionally, the FAQ output could be expanded to include more detailed information, such as menu prices and drinks, to provide users with comprehensive answers to their queries. Finally, the ordering system could be upgraded to accept multiple items per order and display a final price, allowing users to place more complex orders easily. This enhancement would increase the chatbot's utility for customers and improve the overall ordering experience by offering greater flexibility and transparency.

Chatbot Improvements:

In looking towards the future of the Vida Rica chatbot, there are several specific improvements that could significantly enhance its functionality and user experience. First, name input functionality will currently accept any input as a valid input (whether letter, number, or symbol). Ideally this should be improved so that only letter inputs are valid. Second, ideally the chatbot would allow the customer to edit their name if they mistype it. This could be performed by making the name input a separate node such that if a customer gave feedback like "change name" that would trigger an intent to return back to name input. Next, the email input

functionality could be refined to better parse email addresses from any accompanying text. Currently, the chatbot can recognize an email address if entered in the correct format, but users often include additional text, such as "Please contact me at my email, which is example@domain.com." By implementing a parsing feature that extracts the email address from such text, the chatbot could improve accuracy and user satisfaction, minimizing the need for repeated attempts to input valid data.

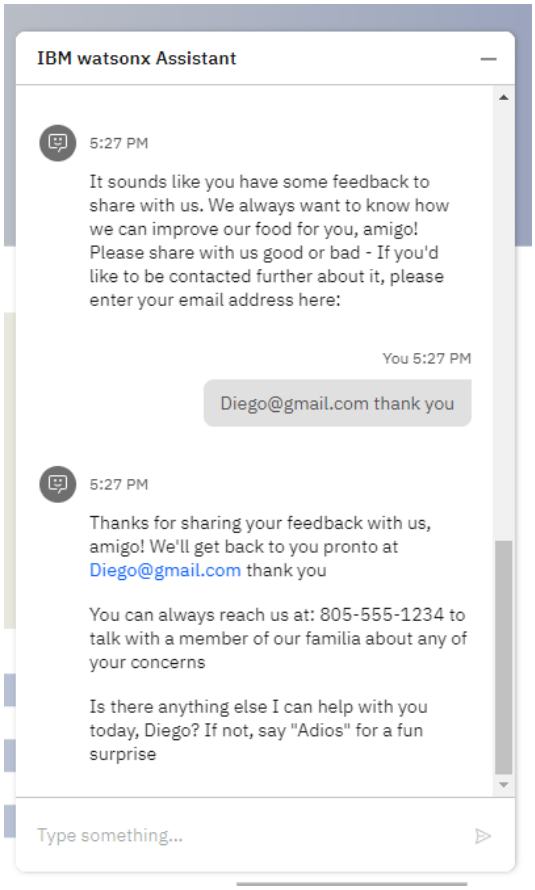

Figure 26. The Watson chatbot interface used by testers showing email input being accompanied along with the “thank you” from the user’s input.

Another potential improvement involves changing how the chatbot handles spiciness level input. Currently, users are asked to enter a percentage to indicate their preferred spice level, which can lead to errors or confusion. By shifting to a more user-friendly option-based system,

where users can select from predefined spiciness levels (e.g., mild, medium, hot, quiero bailar con el diablo), the chatbot would streamline the ordering process and reduce the cognitive load on users (Schmidhuber et al., 2021). This approach aligns with common restaurant practices and enhances the ordering experience by offering clarity and ease. Similarly, the design of the ordering process could be expanded to accommodate multiple items per order, including drinks. This change would involve restructuring the dialog nodes to allow users to add items sequentially, similar to how they would in an online shopping cart. Users could specify the quantity and type of each item and add drinks or sides to their order. This feature would increase order value and customer satisfaction by making it easier to place comprehensive orders. This would also need to be accompanied by a change to how the final order confirmation is performed where instead of cancelling an incorrect order, the bot would allow a user to edit any items in their cart.

In personalizing the interaction with the chatbot, I focused on several key approaches. First, I used context variables to store user information, such as their name, which allowed the chatbot to address users personally throughout the conversation. This personalization helps build rapport and makes the interaction feel more natural. Second, I implemented multimedia elements, including images and videos, to enrich the conversation and create a more engaging environment. These elements along with the use of emojis, a friendly tone, and adding fun elements within the chatbot help convey the brand's personality and make the interaction more enjoyable. Third, I designed the chatbot to use natural language processing techniques, such as recognizing synonyms and varied expressions, to understand user intents better and respond appropriately. There was also coding to greet the user based upon the time of day in order to give context to the user making the bot feel more “alive”. Further personalization in the future could

focus on addressing the person based upon their location, based upon the day of the week, and based upon ensuring that the user is over 21 years of age to buy alcoholic beverages online.

Response disambiguation is a process used to clarify the user's intent when the chatbot is uncertain. For instance, if a user asks, "What are your hours?" and "Where are you located?" in the same query, the chatbot might ask, "Did you mean hours or locations?" to clarify the request. Another example could occur when a user types, "I want a taco and nachos," and the chatbot prompts, "Would you like to add both items to your order?" In the current state, the chatbot has only a low level of response disambiguation where it can parse certain terms but not an extensive amount. For example, if a user attempts to ask for multiple items from the main menu, the bot will offer only the most relevant options from menu based on the input. In contrast, if a user inputs that they want both a taco and nachos, the bot currently is set up to store the first item as the variable to store and it doesn’t store the second item. This is controlled by the chatbot stating from the beginning that only 1 item is allowed per order; but, this obviously is not a long-term solution for most businesses where they want to sell as many items as possible. Again, in an ideal state, the chatbot would be designed to select these items and customizations as buttons/option inputs to minimize cognitive load on the customer (Schmidhuber et al., 2021).

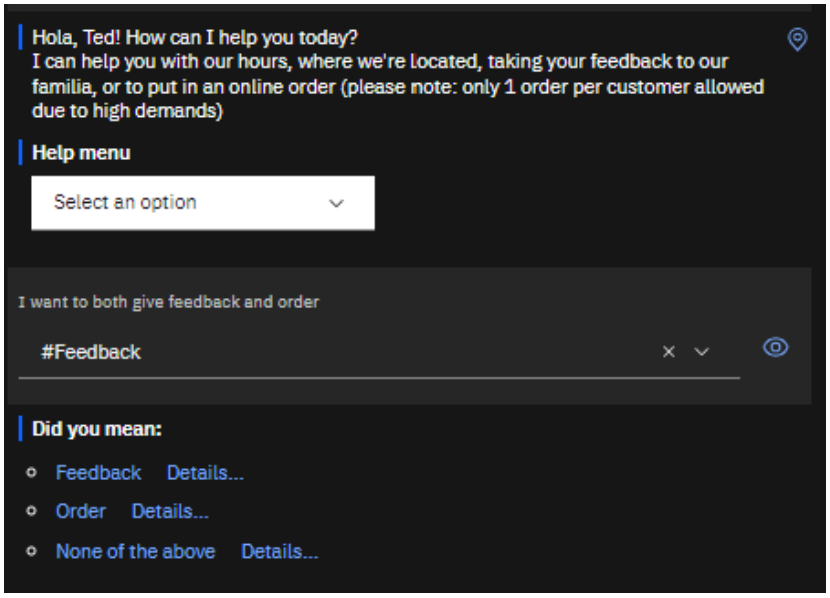

Figure 27. The Watson chatbot interface showing response disambiguation between giving feedback and an order.

To enhance the chatbot's understanding of user input and improve the relevancy of its responses, integrating several Watson services could provide significant benefits. **Natural Language Understanding (NLU)** would allow the chatbot to analyze and interpret user inputs more effectively by extracting entities, sentiment, and intent with greater precision. This would enable the chatbot to understand complex queries and provide more relevant and tailored responses. **Language Translator** could be integrated to support multilingual users, allowing the chatbot to converse in different languages and broaden its accessibility. By automatically translating user inputs and chatbot responses, this service would cater to a more diverse audience, enhancing user satisfaction. This is particularly important to apply to restaurants in Southern California where there is a significant proportion of the population who primarily speak Spanish. This makes the business much more accessible to a wider market of customers. **Text to Speech (TTS)** could provide a voice interaction option, offering a more accessible experience for users who prefer auditory communication or who are unable to type. This would make interactions more engaging and inclusive, especially for visually impaired users. Finally, **Speech**

to Text (STT) could be implemented to convert spoken language into text, allowing users to interact with the chatbot using their voice. This service is particularly beneficial for mobile users or those who find typing cumbersome. Together, these Watson services would significantly elevate the chatbot's capabilities, making it more versatile, user-friendly, and capable of delivering a multilingual, speech/text experience.

Incorporating dynamic content into the chatbot is a challenge, but there are several theoretical approaches a business could consider. One approach is to use external APIs to fetch real-time data, such as current menu items, promotions, or store hours (Hussain et al., 2018). This integration ensures that users receive up-to-date information without requiring manual updates to the chatbot's code. Another approach is implementing a content management system (CMS) where non-technical staff can update chatbot responses easily (Galitsky et al., 2019). This setup allows the chatbot to pull updated content regularly, keeping information fresh and relevant.

When deploying the chatbot, a business like this would have several platforms to choose from. The best usage would include Facebook Messenger, Instagram, and the business' website. The three of these would allow the widest cadre of each generation and of different types of customers to be reached. Messenger allows businesses to reach a large audience easily and supports rich media and interactive elements, enhancing the user experience. It along with Instagram have widespread popularity ensuring that many users are familiar with the interface, reducing friction and encouraging engagement.

To further train the chatbot after deployment, it's essential to adopt a systematic approach that involves continuous monitoring and iterative improvement based on user interactions.

Initially, a business should implement robust analytics to track user interactions and identify patterns where the chatbot might be falling short. By analyzing these patterns, they can pinpoint areas that require updates, such as expanding intent coverage or refining entity recognition to handle diverse inputs more accurately. Incorporating machine learning techniques, the chatbot can be trained on new datasets that reflect evolving language and expressions, ensuring it stays current and responsive to user needs. Moreover, leveraging a mechanism for users to provide direct feedback immediately after interactions can lead to more specific insights, which are crucial for targeted improvements. Encouraging users to rate their experience or report issues can guide iterative enhancements, allowing the chatbot to become more attuned to user needs and preferences over time.

Regular retraining sessions using updated data sets are crucial to refining the chatbot's natural language processing capabilities. This involves not only updating NLP models with new data but also refining underlying algorithms to enhance their ability to parse and understand complex queries. Incorporating advancements in NLP and AI, such as transformers and contextual embeddings, allows the chatbot to achieve higher levels of understanding and contextual awareness (Ahmed et al., 2023). Additionally, collaborating with domain experts and stakeholders to continuously refine the chatbot’s knowledge base ensures that it offers accurate and relevant responses, especially for complex inquiries requiring specialized information. Through these comprehensive training and improvement strategies, the chatbot can evolve into a more powerful and dependable conversational agent. This should help deliver high-quality interactions and drive user satisfaction and engagement.

Conclusion:

The development and refinement of the Vida Rica chatbot highlight the potential of AI-driven solutions to enhance customer service in the hospitality industry. Throughout the process, we focused on creating a user-friendly, efficient, and engaging conversational agent capable of handling a wide range of customer inquiries and interactions. The implementation involved the careful design of user entities and intents to ensure the chatbot could accurately capture and process diverse user inputs. By leveraging Watson Assistant's capabilities, including context variables, slots, and event handlers, the chatbot was able to provide seamless and personalized interactions, leading to a more satisfying user experience. This process showed the creation, testing, implementation, and improvement of a simple chatbot design which has the potential to save hundreds of work hours each month for a business. This type of technology is key for businesses in the hospitality industry to take advantage of in order to meet customer demands and importantly foster brand loyalty.

Several key takeaways emerged from the chatbot development process. First, the importance of understanding user needs and designing intents and entities to capture specific interactions cannot be overstated. The addition of intents like **#FAQ** and **#bridge** allowed the chatbot to handle frequently asked questions and assist users who needed additional guidance. Second, the use of regular expressions for input validation proved crucial in ensuring the quality and accuracy of the data collected. By validating inputs such as email addresses and phone numbers, the chatbot was able to reduce errors and enhance the reliability of interactions. Crucially, the inclusion of multimedia and human elements like images, emojis, jokes, and videos added an engaging layer to the chatbot, making interactions more enjoyable and visually

appealing. And finally, using the customer's input name throughout the interaction makes the bot feel more friendly and human to users.

Despite the successful implementation, several challenges were encountered during the development of the chatbot. One challenge was ensuring that the chatbot could understand and respond to a wide variety of user expressions and synonyms, which required extensive training and refinement of intents and entities. From the original design, to after day 1 of testing, to day 2 of testing, each iteration required further improvement to make the bot more intelligent and operative. Another challenge was implementing effective input validation to ensure only accurate data was processed, which involved complex logic and regular expressions. Finally, maintaining a smooth and cohesive conversation flow while handling unexpected inputs or errors required careful design of dialog nodes and event handlers.

To address these challenges and build "smart" chatbots, several recommendations can be made. First, it's critical to invest in continuous training and refinement of intents and entities to improve the chatbot's understanding of user inputs. This can be achieved through regular analysis of interaction logs and updating the chatbot with new examples and synonyms. Second, implement robust input validation and error handling mechanisms to ensure data integrity and provide users with clear feedback and corrective prompts when necessary. Third, enhance the chatbot with natural language processing capabilities to better understand and respond to complex queries and provide more human-like interactions. Bouguezzi (2024) gives some extensive explanation in regard to this topic. In developing a smart chatbot, integrating NLP and AI is crucial to create a system that intuitively understands and responds to user intents, much like a human would. The Vida Rica chatbot would ideally utilize advanced NLP techniques to enhance its conversational abilities beyond the limitations of rule-based systems. By

implementing NLP, the chatbot can parse and interpret user inputs through processes such as tokenization, normalization, and entity recognition, which enable it to understand the context and intent behind each message (Bouguezzi, 2024). This understanding allows the chatbot to classify user intents accurately and generate responses that are both relevant and coherent within the given context. Unlike rule-based chatbots that rely on predefined instructions, a smart chatbot with NLP capabilities can dynamically adapt to new inputs, generating unique responses that reflect a deeper understanding of user interactions. Such sophistication in AI implementation ensures that the chatbot can engage in meaningful dialogues, handle complex queries, and provide a more personalized experience for users.

While the Vida Rica chatbot offers a high level of service automation, it is unlikely to fully replace human interactions. Even with an ideally complex chatbot using advanced NLP techniques, chatbots will never fully replace humans. There are several reasons for this. First, some customer inquiries and issues may require nuanced understanding and empathy that only a human can provide. Second, complex or unusual requests might be beyond the chatbot's capabilities and require human intervention. Finally, some customers may simply prefer speaking with a live person, especially for sensitive or detailed inquiries. Therefore, the chatbot should be viewed as a complement to human service, handling routine and repetitive tasks while freeing up human agents to focus on more complex and personalized interactions. This hybrid approach ensures that customers receive the best of both worlds: efficient automated service and personalized human touch when needed. But the number of hours returned back to the business by implementing a simple chatbot like this is astronomical and can result in significant profit for the company. Lastly and importantly, a chatbot like Vida Rica also helps to foster a stellar brand

where the business' personality can be displayed through logos, humor, personal touch, top notch customer service, and warmth helping to retain customers.

In conclusion, chatbots like this are critical for the hospitality industry to adopt. This analysis has shown in total: the importance of chatbots, the design process of a chatbot, the implementation of a chatbot into a website where simulated customers used it, and the improvement process of a chatbot addressing areas where the bot was not able to handle messages and adding new nodes into it to make the user experience more efficient and friendly thus promoting organizational productivity and customer loyalty.

References:

Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 373-383). Springer, Cham.

Ahmed, M., Khan, H. U., & Munir, E. U. (2023). Conversational ai: An explication of few-shot learning problem in transformers-based chatbot systems. *IEEE Transactions on Computational Social Systems*, *11*(2), 1888-1906.

Bouguezzi, S. (2024). How to Create a Smart Chatbot? Baeldung. https://www.baeldung.com/cs/smart-chatbots

Bratić, D., Šapina, M., Jurečić, D., & Žiljak Gršić, J. (2024). Centralized database access: transformer framework and llm/chatbot integration-based hybrid model. Applied System Innovation, 7(1), 17.

Dash, M., & Bakshi, S. (2019). An exploratory study of customer perceptions of usage of chatbots in the hospitality industry. *International Journal on Customer Relations*, *7*(2), 27-33.

Galitsky, B., & Galitsky, B. (2019). A content management system for chatbots. *Developing Enterprise Chatbots: Learning Linguistic Structures*, 253-326.

Hussain, S., & Athula, G. (2018, May). Extending a conventional chatbot knowledge base to external knowledge source and introducing user based sessions for diabetes education. In *2018 32nd international conference on advanced information networking and applications workshops (WAINA)* (pp. 698-703). IEEE.

IBM. (2022). Classic Watson Assistant Product Guide. IBM. https://cloud.ibm.com/docs/assistant?topic=assistant-index

Mizuguchi, S. (2017). Parsing email and phone number entities? StackOverflow. https://stackoverflow.com/questions/45092787/parsing-email-and-phone-number-entities

Schmidhuber, J., Schlögl, S., & Ploder, C. (2021). Cognitive load and productivity implications in human-chatbot interaction. In *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)* (pp. 1-6). IEEE.

Ursu, S. (2019). How to Design a Chatbot Personality. Medium. https://medium.com/the-chatbot-guru/how-to-design-a-chatbot-personality-8f89b5a6b549

Attachments:

Attachment 1: Defined Terms

Index	Word	Definition
1	Intent	The purpose or goal behind a user's input. It represents the action the user wants to perform. **Example:** Booking a flight, ordering food.
2	Entity	Specific pieces of information in the user's input that help to clarify their intent. Entities can be user-defined (my entity) or system-defined (system entity).
3	My entity values	Custom values defined within a user entity. **Value synonyms:** Different words or phrases that mean the same value. **Value patterns:** Patterns to match values.
4	Dialog skill	A set of dialog nodes used to manage conversations in the chatbot.
5	Search skill	Allows the chatbot to retrieve information from external sources based on user queries.
6	Dialog Node	The building blocks of a conversation in a dialog skill. Properties include conditions, responses, and context variables.
7	Dialog node Entry condition	Conditions that must be met for a dialog node to be triggered.
8	Welcome node	The initial node that is triggered when a conversation starts.
9	Anything else node	A fallback node that is triggered when none of the other nodes match the user input.
10	Child node	A sub-node under a parent node, used to handle specific branches of a conversation.
11	Parent node	A main node that contains child nodes, helping to structure complex dialogs.
12	Response types	The different formats of responses the chatbot can give: **Text:** Plain text. **Image:** Visual content. **Option:** List of options. **Search:** Retrieved info.

Index	Word	Definition
13	Jump to	An action that allows the conversation to move to another dialog node based on conditions or responses in the target node.
14	Slot	Variables within a dialog node that can be filled with user input to capture specific information.
15	Handler	Logic within a dialog node that processes user input and determines the next action.
16	Context variable	Variables that store information about the conversation state, which can be used across multiple dialog nodes.
17	Conditional response	Responses that vary based on specific conditions or context variables.
18	Response variation	Different versions of a response to avoid repetition and make interactions feel more natural.
19	Folder	A way to organize dialog nodes into groups for better management and navigation.
20	Try it out window	A testing interface within the chatbot design environment to simulate and test conversations.
21	Assistant	The overall AI entity that interacts with users, composed of dialog and search skills.
22	Conversation Analytics	Tools and metrics used to analyze the performance and effectiveness of the chatbot interactions.
23	Chatbot Preview link	A link to preview the chatbot's functionality before deployment.
24	JSON file	A file format used to export and import the configuration of the chatbot, including intents, entities, and dialog nodes.
25	Conversation scenario	A predefined set of interactions designed to test and demonstrate the chatbot's capabilities.

Figure 28. Chatbot and Watson-related vocabulary needed in order to understand this analysis and to build/implement chatbots (terms fetched from IBM (2022)).

Attachment 2: Testing Instruction for Users

Vida Rica! Chatbot Testing Instructions (03AUG24):

Purpose:

This chatbot was built as part of an assignment for DATA 650 from UMGC. Thanks for taking the time to test it out – it really helps and is a pivotal part of this assignment to make it better.

Vida Rica! is a fictional Mexican restaurant I made up for the purpose of this assignment.

This chatbot is pretty simple in its design – it won't know anything about quantum mechanics, but it should be able to answer some typical questions people would have about a takeout Mexican restaurant (basics like hours or locations, not attire or if they have anything gluten free).

Instructions:

You will need to test the chatbot over 2 days: Saturday 03AUG, and Sunday 04AUG.

Access the bot here:

https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?region=us-south&integrationID=6f134cad-d0b6-4193-b468-c1c745cdb028&serviceInstanceID=0a4e204b-3f46-45c9-b354-09d06cff91d6

You will need to run 3-4 conversations with the bot. You'll need to remember these conversations to have the same ones on the second day that you had on the first day (so please keep track).

Typical conversation flow:

Greetings → Questions/Select from dropdown menu → More questions if needed → Goodbye

The bot works via detecting keywords so you should be able to type in a keyword for it to detect anytime.

If you have any questions, please let me know. If you see the chatbot do anything weird or unexpected, feel free to let me know as well.