Chatbot Deployment with IBM Cloud Watson Assistant-phase 5

CHATBOT DEPLOYMENT WITH IBM CLOUD WATSONASSISTANT

PROBLEMDEFINITION TheprojectinvolvescreatingachatbotusingIBMCloudWatsonAssistant.Thegoalistode velopavirtualguidethatassistsusersonmessagingplatformslikeFacebookMessengera ndSlack.

The chatbot should provide helpful information, answer frequently asked questions (FAQs), andofferafriendlyconversational experience.Theprojectincludesdesigningthechatbot'spersona, configuringresponses,integratingwithmessagingplatforms,andensuringaseamlessuse rexperience.

APPROACHMETHOD

ForFacebook: **Facebookdevelopersplatform:**

•

GototheFacebookforDevelopersplatfor[m(https://developers.fac](https://developers.facebook.com/) [ebook .com/).](https://developers.facebook.com/)

• CreateanewFacebookApporuseanexistingone.

**SetUpMessengerIntegration**:

• InsideyourFacebookApp,navigatetotheMessengertabandsetuptheMes sengerintegration.

• GenerateaPageAccessToken,whichisneededtointeractwith FacebookMessenger.

**ConfigureWebhooks:**

• SetupawebhooktoreceiveincomingmessagesfromFacebookMessen ger.Thiswebhookshouldpointtoyourserverorcloudservice.

**DevelopaBackendServer:**

• CreateabackendserverthatcommunicateswithWatsonAssistant.Thiss erverwillreceive messages from Facebook, send them to Watson Assistant, and relay theresponsesbacktoFacebook.

**IntegratewithWatsonAssistant:**

• UsetheWatsonAssistantAPI tointeractwithyourchatbot.SendusermessagestoWatsonAssistantan dreceiveresponses.

**HandleUserAuthentication(if necessary):**

• Dependingonyourusecase,youmightneedtohandleuserauthenticati onwithinyourchatbot.

**TestandDeploy:**

• Testyourintegrationthoroughlytoensureitworksasexpected. • Deployyourintegrationtoaproductionenvironment.

**ConnecttoFacebookApp:**

• IntheFacebookAppsettings,linkyourbackendserver(webhook)totheMe ssengerintegration.

**SubmitforReview:**

• Ifyouplantomakeyourchatbotavailabletothepublic,you'llneedtosu bmitittoFacebookforreviewandapproval.

ForSlack: **CreateaSlackApp:**

• GototheSlackAPI websit[e(https://api.slack.com/).](https://api.slack.com/) • CreateanewSlackApporuseanexistingone.

**ConfigureBotUser:**

• InsideyourSlackApp,configureabotuser.Thisbotuserwillrepresentyour WatsonAssistantchatbot.

**InstalltheApptoWorkspaces:**

• Enableyour Slack App to be installed by Slack usersin their workspaces.

**ImplementOAuth2.0:**

• SetupOAuth2.0authenticationtoallowyourapptoaccessSlackworkspace s.

**IntegratewithWatsonAssistant:**

• UsetheWatsonAssistantAPItosendandreceivemessagesfromSlack.

**DevelopaBackendServer:**

• Createabackendserverthatlistensforeventsandmessagesfrom SlackandcommunicateswithWatsonAssistant.

**TestandDeploy:**

• Testyourintegrationandensureithandlesvarioususerinteractions. • Deployyourintegrationtoaproductionenvironment.

**ConnecttoSlackApp:**

• IntheSlackAppsettings,configureeventsubscriptionsandinteractiveco mponentstohandleincomingmessages.

**SubmitforDistribution:**

• IfyouwanttomakeyourchatbotavailabletootherSlackworkspaces,su bmityourappfordistributionthroughtheSlackAppDirectory.

DESIGNTHINKING

**PersonaDesign:** •

Creatingachatbotpersonahelpsdefineitscharacter,whichinfluencesh owusersperceiveandinteractwithit.Itsetsthetoneandstyleofcommu nication,makingitrelatabletousers.

**UserScenarios:**

• Identifying common user scenarios and FAQs ensures that the chatbot can effectivelyaddress the needs and queries of its intended audience. This step helps in building achatbotthatprovidesvalue.

**ConversationFlow:**

• Designingtheconversationflowoutlineshowthechatbotrespondstodiff erentuserinputs. It's crucial for guiding the chatbot's interactions and ensuring a coherent andhelpfuldialogue.

**ResponseConfiguration:**

• LeveragingWatsonAssistant'sfeatureslikeintents,entities,anddialognodesi svitalfor configuringthechatbot'sresponses.Thisstepenablesthechatbottoundersta ndandgenerateappropriaterepliesbasedonuserinputs.

**PlatformIntegration:**

• IntegratethechatbotwithpopularmessagingplatformslikeFacebookMess engerandSlack.

• Integratingthechatbotwithpopularmessagingplatformsextendsitsreachand

accessibility.

**UserExperience:**

• Prioritizingaseamlessanduser-friendlyexperienceiskey.Clearpromptsandinformativeresponsesenhanceus ersatisfactionandthechatbot'seffectiveness.

**DESIGN THINKING:**

**Persona Design:**

⁃ Creating a chatbot persona helps define its character, which influences how users perceive and interact with it. It sets the tone and style of communication, making it relatable to users.

**User Scenarios:**

⁃ Identifying common user scenarios and FAQs ensures that the chatbot can effectively address the needs and queries of its intended audience. This step helps in building a chatbot that provides value.

**Conversation Flow:**

⁃ Designing the conversation flow outlines how the chatbot responds to different user inputs. It's crucial for guiding the chatbot's interactions and ensuring a coherent and helpful dialogue.

**Response Configuration:**

⁃ Leveraging Watson Assistant's features like intents, entities, and dialog nodes is vital for configuring the chatbot's responses. This step enables the chatbot to understand and generate appropriate replies based on user inputs.

**Platform Integration:**

⁃ and Slack.

⁃ accessibility.

Integrate the chatbot with popular messaging platforms like Facebook Messenger

Integrating the chatbot with popular messaging platforms extends its reach and

**User Experience:**

⁃ Prioritizing a seamless and user-friendly experience is key. Clear prompts and informative responses enhance user satisfaction and the chatbot's effectiveness

INNOVATION

**1. User Input**: The chatbot begins by receiving user input, which can be in the form of text . Gather a diverse dataset of user inputs and intents to train your chatbot. Use this data to train your NLU model for better intent recognition.

**2. Natural Language Understanding (NLU**): Watson's NLU component analyzes the user input to understand the intent and entities mentioned in the message. This step involves techniques like tokenization, entity recognition, and sentiment analysis.

**3. Intent Recognition**: Based on the NLU analysis, the chatbot determines the user's intent. For example, if the user asks about the weather, the intent is to inquire about the weather forecast.

**4. Entity Recognition**: Define entities that your chatbot needs to understand, such as dates, locations, products, etc. Train the NLU model to recognize and extract these entities from user inputs.

**5. Dialog Management**: The chatbot uses a dialog management system to maintain context and decide how to respond. This includes keeping track of the conversation history and deciding whether to ask clarifying questions or provide a direct response.

**6. Response Generation**: Using the identified intent and extracted entities, the chatbot generates a response. This response can be in the form of text, voice, or other media, depending on the application.

**7. User Interaction**: The chatbot sends the response to the user and waits for further input. The user can continue the conversation or ask new questions.

**8. Fallback Mechanism**: If the chatbot cannot understand the user's input or intent, it may trigger a fallback mechanism to handle unexpected or unclear queries

**9.Context Management**: Implement context management to maintain the conversation context over multiple turns. This helps in understanding user queries that depend on previous interactions.

**10. Natural Language Understanding features:**

***Sentiment Analysis***: NLU can determine the sentiment or emotional tone of user messages, which is valuable for understanding user satisfaction or dissatisfaction

***Language Variations***: NLU models should be able to understand different languages, dialects, and regional variations to accommodate a diverse user base.

***Error Handling***: NLU can recognize and handle user errors gracefully, providing helpful suggestions orcorrective actions.

***Emotion Detection***: Detecting user emotions can help tailor responses accordingly, providing empathy and support when needed.

**11. Integration with Watson Assistant**: Integrate the NLU capabilities into your Watson Assistant chatbot. This involves passing user messages through the NLU model before processing them in the chatbot.

**12. Continuous Learning**: Regularly update and retrain your NLU model with new data to adapt to changing user behaviors and language patterns.

**13.Testing and Evaluation**:Thoroughly test your chatbot with various user inputs to ensure accurate intent recognition and entity extraction. Collect user feedback and make improvements based on real-world interactions.

**14.Monitoring and Analytics**: Implement monitoring tools to track the chatbot's performance, analyze user interactions, and identify areas for improvement.

**15. Analytics and Learning**: Watson chatbots often collect data on user interactions to improve their performance over time. This data can be used for analytics and to train the chatbot to better understand user intents.

**CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT**

Creating a Watson Assistant involves several steps. Here's a general overview:

**Sign Up for IBM Cloud:**

If you don't have an IBM Cloud account, you'll need to sign up for one at [IBM](https://cloud.ibm.com/) [Cloud.](https://cloud.ibm.com/)

**Access Watson Assistant:**

After signing in, go to the IBM Cloud dashboard.

**Create a Watson Assistant Service**:

Click "Create Resource."

Search for "Watson Assistant" in the catalog and select it. Choose the pricing plan that fits your needs.

**Service Configuration:**

Give your service a name.

Choose the region for your service.

Click "Create" to provision the Watson Assistant service.

**Create an Assistant:**

Once the service is provisioned, click on it.

Under "Manage," click "Launch Watson Assistant."

**Create a Skill:**

Inside Watson Assistant, create a new skill. A skill represents a chatbot or virtual assistant.

Configure your skill by defining intents, entities, and dialog flows.

**Train Your Assistant:**

Teach your assistant by adding examples for each intent and setting up dialog nodes forconversatons.

**Integration:**

Decide how you want to integrate your assistant. You can integrate it into a website, mobile app, or other platforms.

**Test and Refine:**

Test your assistant to make sure it understands user queries and responds accurately. Refine your training data and dialog as needed.

**Deploy Your Assistant:**

Once you're satisfied with your assistant's performance, deploy it to the platform where you want it to be used.

**Continuous Improvement:**

Regularly monitor the performance of your assistant and gather user feedback to improve it over time.

**Log in to IBM Cloud:**

Go to the IBM Cloud website and log in to your account.

**Navigate to Resource Groups:**

Once logged in, click on the "Resources" section in the IBM Cloud Console.

**Create a New Resource Group:**

In the Resource Groups section, you should find an option to create a new resource group. Click on it.

**Actions:**

Actions are used to define custom logic or to trigger specific behaviors in response to user input. They can involve setting context variables, generating responses, or making external API calls.

Actions are custom code or logic that the chatbot can execute in response to user input. This might involve making API calls, modifying context variables, or other custom behavior.

*EXAMPLE CONVERSATION OF CHATBOT IN FACEBOOK MESSENGER*

**User:**

Hi, I need to change my Facebook password. Can you guide me through the process?

**Chatbot:**

Absolutely, I can help with that. To change your password, you'll need to follow these steps:

 Open the Facebook app on your device.

 Tap the three horizontal lines in the bottom-right corner to access the menu.  Scroll down and select "Settings & Privacy."

 Tap "Settings."

 Scroll down to the "Security" section and tap "Security and Login."

 Under "Login," you'll find an option to "Change password." Tap on it. **User:**

Got it, I've reached the "Change password" option.

**Chatbot:**

Great! Now, you'll be prompted to enter your current password for verification. Please enter your current password.

**User:**

Password entered.

**Chatbot:**

Perfect. Now, you can set your new password. Be sure to choose a strong and unique one. Once you've entered it, confirm it by typing it again.

**User:**

New password set and confirmed. **Chatbot:**

Fantastic! Your password has been changed successfully. You should receive a confirmation message. Is there anything else you'd like assistance with?

**User**:

No, that's it. Thanks for your help!

**Chatbot:**

You're welcome! If you have any more questions in the future, don't hesitate to reach out. Have a great day!

Intents

In Watson Assistant, "intents" are a fundamental concept used to understand the meaning or purpose behind a user's input. They help the chatbot recognize what a user is trying to accomplish or the main topic of their message. Here's how intents work in Watson Assistant:

**Intent Recognition:**

When a user sends a message to the chatbot, Watson Assistant's natural language processing capabilities analyze the text to identify the user's intent. This intent recognition is based on the patterns and keywords present in the user's message.

**Intent Definitions:**

You, as a developer, define a set of intents that are relevant to your chatbot's domain or purpose. For example, in a weather chatbot, you might define intents like "get\_weather," "check\_forecast," "ask\_temperature," etc.

**Training Data:**

To train the chatbot to recognize these intents, you provide examples of user messages for each intent. These examples are used to teach Watson Assistant's machine learning models to associate certain phrases and patterns with specific intents.

**Intent Confidence:**

For each user message, Watson Assistant not only identifies the intent but also provides a confidence score. This score indicates how certain the chatbot is that it correctly recognized the user's intent. You can use this confidence score to make decisions in your dialogue nodes.

**Dialog Flow:**

Once the intent is recognized, you can use it to guide the conversation flow. Depending on the detected intent, you can route the conversation to different dialogue nodes or provide appropriate responses.

**Fallback Intent:**

You can define a "fallback" intent to handle situations where the chatbot cannot confidently identify any other intent. This allows you to gracefully respond to user messages that are outside the scope of defined intents.

**Context Management:**

Intents often work in conjunction with entities and context variables to provide context-aware responses and gather necessary information from the user.

Entities

**1. Recognition of Structured Information:**

Entities are used to identify and extract structured data from user messages. This structured data can be specific details such as dates, times, numbers, locations, product names, or any other relevant information.

**2. Entity Types:**

Watson Assistant supports various entity types, including system entities and user-defined entities. System entities are predefined for common concepts like numbers, dates, and times, while user-defined entities are created to match domain-specific data.

**3. Training Data:**

To teach the chatbot to recognize entities, you provide training data that includes examples of user messages containing the entities you want to extract. For instance, if you're building a chatbot for ordering pizza, you might create an entity called "PizzaTopping" and provide training examples like "I want a pizza with pepperoni."

**4. Entity Values:**

Within an entity type, you define entity values. For example, if you have an entity type "PizzaTopping," the entity values could be "Pepperoni," "Mushroom," and "Onion." These values represent the specific options that the chatbot should recognize within the entity.

**5. Entity Matching:**

When a user sends a message to the chatbot, the platform uses natural language processing to analyze the message and identify which entity values are present. For instance, if a user says, "I'd like a pizza with pepperoni and mushrooms," the chatbot might recognize the "PizzaTopping" entity with the values "Pepperoni" and "Mushroom."

**6. Contextual Data:**

Extracted entities are used to provide context and assist the chatbot in generating relevant responses. In the pizza example, the chatbot can use the identified pizza toppings to accurately fulfill the user's order.

**7. Multiple Entities:**

In a single message, users may mention multiple entities. Watson Assistant is designed to identify and extract multiple entities to enhance its understanding of user input.

Dialogue

In Watson Assistant, dialogue nodes are essential components that define the flow and interaction of a chatbot with users. They represent specific points in the conversation where the chatbot processes user input and provides responses. Here's how dialogue nodes work:

**Trigger Condition:**

Each dialogue node typically has a trigger condition that specifies when the node

should be activated. For example, a trigger condition might be based on specific user

input or context.

**User Input:**

When the trigger condition is met, the chatbot processes the user's input provided up

to that point in the conversation. It may use natural language processing to understand

the user's intent and entities mentioned.

**Response:**

The dialogue node contains a response that the chatbot will deliver to the user based

on their input. This response can be static text or dynamically generated based on

variables or logic.

**Actions:**

Dialogue nodes can also include actions, such as setting context variables, invoking

external services, or branching to other nodes. These actions help control the conversation flow.

**Conditions:**

Conditional logic can be used in dialogue nodes to guide the conversation. For example, you can create conditions based on user input or variables to direct the chatbot's responses.

**Slots:**

In more complex interactions, slots can be used to gather specific information from the

user, like dates, numbers, or choices.

**Digressions:**

Watson Assistant supports various digressions, allowing the chatbot to handle multiple

topics simultaneously or return to the main conversation flow after addressing a side

topics.

INTEGRATING CHATBOT WITH MESSAGING PLATFORM

**1. Create a Watson Assistant Chatbot:**

**Start by creating a chatbot using IBM Watson Assistant. Define the intents, entities, and dialog flows that your chatbot will handle.**

**2. Choose a Messaging Platform:**

**Select the messaging platform where you want to integrate the chatbot. Common options include Facebook Messenger, Slack, or a custom web chat.**

**3. Integration Method:**

**Depending on the platform, you may need to use different integration methods. Common approaches include using APIs, webhooks, or SDKs provided by the messaging platform. Watson Assistant also provides a webhook integration option.**

**4. Develop the Integration:**

**Write code or scripts to connect the messaging platform to Watson Assistant. This often involves setting up a middleware that handles user messages, sends them to the chatbot, and relays the responses back to the user.**

**5. Authentication and Authorization:**

**Ensure you have the necessary credentials and permissions to access both the messaging platform and Watson Assistant.**

**6. Message Handling:**

**Implement logic to parse incoming messages, understand user intent, and send the appropriate queries to Watson Assistant.**

**7. Response Processing:**

**Handle responses from the chatbot and format them for display in the messaging platform.**

**8. Testing:**

**Thoroughly test the integration to make sure the chatbot functions correctly within the messaging platform. Test various user inputs and dialog flows.**

**9. Deployment:**

**Deploy the integrated chatbot on the messaging platform for users to interact with.**

**10. Monitoring and Maintenance:**

**Continuously monitor the chatbot's performance, gather user feedback, and make improvements to the chatbot's responses and capabilities.**

CHATBOT ON FACEBOOK **1. Create a Watson Assistant Chatbot:**

**Start by creating a chatbot using IBM Watson Assistant. Define the intents, entities, and dialog flows that your chatbot will handle.**

**2. Set Up a Facebook Page:**

**If you don't already have one, create a Facebook Page for your business or organization. This is where your chatbot will interact with users.**

**3. Create a Facebook App:**

**You'll need to create a Facebook App to access the Messenger API. Go to the Facebook for Developers website, create a new app, and configure the app settings.**

**4. Get Page Access Token:**

**Within your Facebook App settings, obtain a Page Access Token. This token will allow your app to send and receive messages on behalf of your Facebook Page.**

**5. Set Up Webhooks:**

**Configure the webhook for your Facebook App. This is where Facebook will send incoming messages to your chatbot. You'll need to specify a callback URL for your server where you'll handle incoming messages.**

**6. Develop the Integration:**

**Write code or scripts to connect your Watson Assistant chatbot to Facebook. You'll need a server to handle the incoming messages and send them to Watson Assistant. Use the Page Access Token to send responses back to Facebook Messenger.**

**7. Authentication and Authorization:**

**Ensure you have the necessary credentials and permissions to access both Facebook and Watson Assistant.**

**8. Message Handling:**

**Implement logic to parse incoming messages from Facebook, understand user intent, and send the appropriate queries to Watson Assistant.**

**9. Response Processing:**

**Handle responses from Watson Assistant and format them for display in Facebook Messenger.**

**10. Deployment:**

**Deploy the integrated chatbot on your Facebook Page for users to interact with.**

Thank you