CHATBOT DEPLOYMENT WITH IBM CLOUD WATSON ASSISTANT

PROBLEM DEFINITION AND DESIGN THINKING:

PROBLEM DEFINITION:

The problem is to deploy a chatbot using IBM cloud Watson assistant to address specific business or customer service needs.Our project is solution which is customized for specific requirements for marketing agencies.The marketing agencies are facing lot of challenges including lead generation,client communication and finally reporting.Our project helps to overcome all these challenges by enhancing client interactions and improving overall agency efficiency.It also enhances client communication by providing real-time updates on project progress and addressing common queries.The main objective is to give better experience with defined personas with its name and the tone,style of the chatbot.It is informative and can give endless service to the users and can have friendly conversation to the users .

SOLUTION:

Chatbot deployment using IBM Cloud Watson Assistant involves several steps. First, the chatbot is designed and built using the IBM Cloud Watson Assistant platform. This includes defining intents, creating dialog flows, and training the chatbot using sample user inputs. Once the chatbot is developed, it is integrated into any desired channels, such as websites, mobile apps, or messaging platforms. Watson Assistant provides easy integration options and supports various communication channels. After integration, the chatbot is thoroughly tested to ensure its functionality and accuracy. This involves testing different user scenarios, evaluating the chatbot's responses, and making necessary adjustments to improve its performance. Once the testing is complete, the chatbot is deployed to the live environment. Watson Assistant offers deployment options that allow you to scale and manage the chatbot effectively. After deployment, continuous monitoring and maintenance are crucial. Watson Assistant provides analytics and insights to track user interactions, identify areas for improvement, and enhance the chatbot's performance over time. In summary, chatbot deployment using IBM Cloud Watson Assistant involves designing, building, integrating, testing, deploying, and monitoring the chatbot to provide effective and efficient conversational experiences for users.

DESIGN THINKING:

PERSONA DESIGN:

- ❖ A chatbot persona is the soul of a chatbot, carefully crafted that embodies the tone, voice and personality of virtual assistant.

- ❖ They help in enhancing user experience, establishing brand identity, and building trust and engagement with users.

- ❖ To create a persona we need to do a lot of research about customer needs. By making assumptions and web analytics we can get more datas to create a persona.

- ❖ A chatbot persona is created by following the process involving Identifying user demographics, Aligning with brand values, Developing chatbot's tone and voice and Creating a visual representation.

USER SCENARIO:

- ❖ A chatbot is useful to automatically answer Frequently Asked Questions(FAQs) and reduce the number of messages to your support team.

- ❖ The chatbot can also help to know more about visitors and their needs before transferring them to the support team if needed.

- ❖ Once set up, the chatbot automatically sends pre-configured reply options called scenarios to visitors depending on the webpage they are visiting.

- From the chatbot, visitors choose one of the preconfigured reply options and the bot will send them the next automatic message until the scenario ends or visitors choose to talk to an agent**.**

  These scenarios are activated through triggers.

CONVERSATION FLOW:

- It prompts the user for the user's input or queries. Based on the input it generates response that might be accurate or precise, for complex queries it extracts the data from the database and sends response.

- Specially it can also incorporate the multimedia elements like images, videos into the conversation.
- Finally it collects the user feedback to improve the chatbot's performance and usability.

RESPONSE CONFIGURATION:

- Chatbot uses Natural Language Processing (NLP) to recognize user's intent based on the input given. It creates dialog nodes to define the conversation flow.

- Each node in the dialog nodes will be associated with an intent, condition or context variable.

- After analyzing the inputs, chatbot sends response in the form of text, audio, video, reference links inorder to improve user friendly conversation.
- Default responses can also be set for unrecognized or ambiguous queries.

PLATFORM INTEGRATION:

- Chatbot Integration allows the software to access product catalogs from the e-stores based on eCommerce platforms. It is easier than ever with Watson Assistant.

- Seamlessly connect the customer engagement channels with the CRM systems, search tools, data sources, and contact center platforms that power up business without migrating tech stack.
- Streamline workflow and customer experience to boost customer satisfaction.

- Chatbot involves various Platform Integrations such as Channel, Messaging, LiveAgent and Telephony and Search Integrations. These integrations results in Chatbot Deployment on websites and mobile apps.

USER EXPERIENCE:

- A chatbot can be designed either within the constraints of an existing platform or from scratch for a website or app.
- In order for a chatbot to be well-received, its intended users must be thoroughly researched so the designer can give it an appropriate personality.
- Chatbot User Experience provides user-friendly experience and instant responses.
- The UX Design Process involves simplified design process and emphasize human interaction by setting up communication channels with chatbots integrated with chatbot platform.
- It also focus on the features and functionalities to improve customer experience.

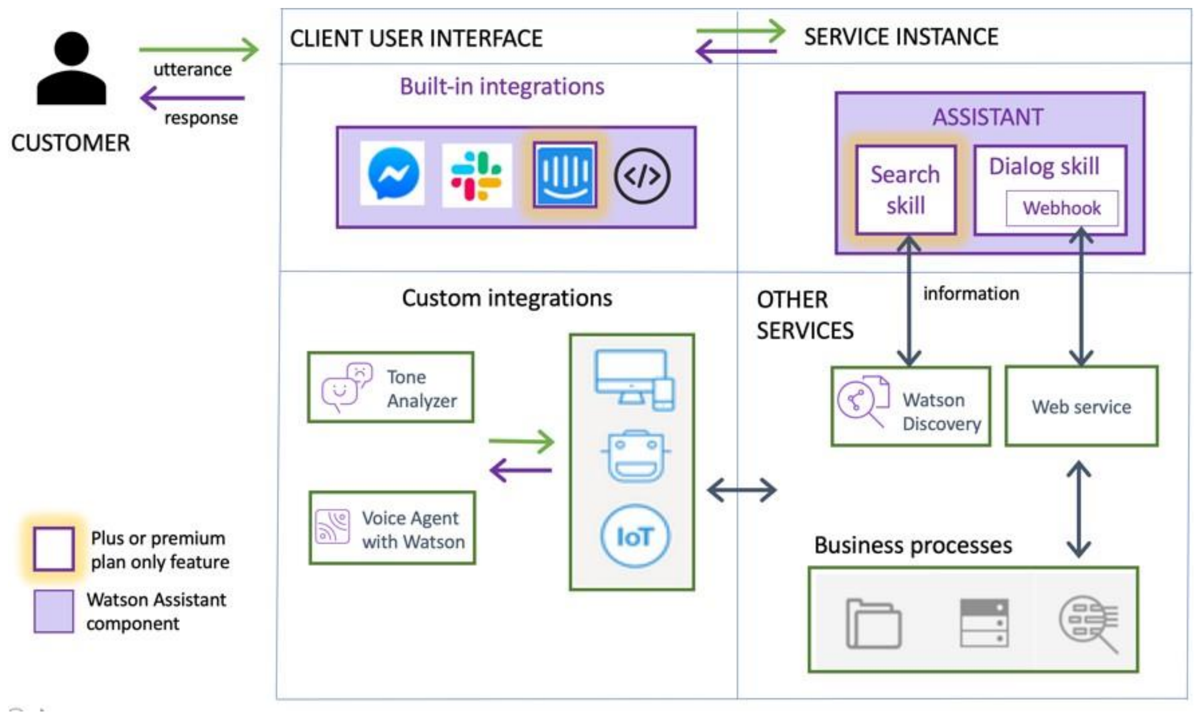
INNOVATION:

OVERVIEW:

Over the last few years, chatbots have become more widely available and more popular as a customer service solution. Chatbot deployment involves the process of implementing and launching a chatbot into a live environment to interact with users. They can be deployed on various platforms, such as websites, mobile apps, messaging platforms, or voice assistants, and assist users by answering questions, providing information, guiding through processes, or even performing tasks like making reservations or placing orders.

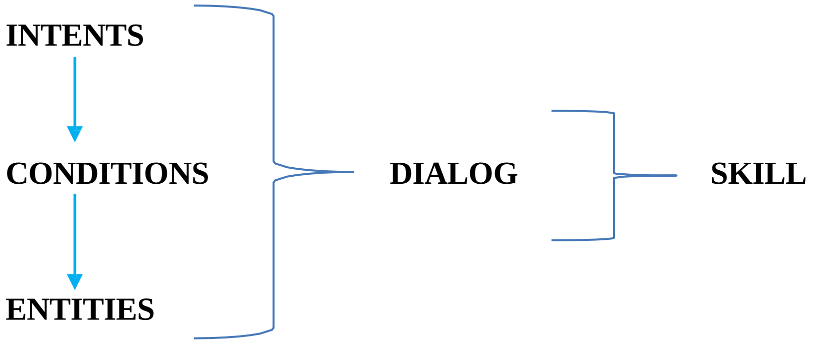
Our project is a solution that is customized for specific requirements for marketing agencies. The marketing agencies are facing a lot of challenges, including lead generation,client communication, and finally reporting. Our project helps to overcome all these challenges by enhancing client interactions and improving overall agency efficiency. It also enhances client communication by providing real-time updates on project progress and addressing common queries. The main objective is to give a better experience with defined personas with the name, tone, and style of the chatbot. It is informative, can give endless service to the users, and can have friendly conversations with them.

WORKFLOW OF CHATBOT AT THE BACKEND:



PROCESS:

Steps to be followed to deploy a chatbot :



Intents: Intent in the context of a chatbot refers to the user's purpose or goal when they interact with the chatbot**.**

Entities: Entities in a chatbot refer to specific pieces of information or data that the bot needs to extract from the user's input in order to understand and respond accurately.

Step 1: Go t**o** IBM **cloud's** login page and create an account to access the the resources that is present in it.

Step2: Next click on the **catalog** menu which is present on the left most corner of the web page.

Step 3: Click and move forward to **Watson assistant** tab.

Step 4: Move on to **Resource list** menu and slide towards to **Services** menu item to access other services from IBM Watson assistant.

Step 5: Press **launch Watson assistant** button which will be displayed in blue.From step 5 actual deployment of chatbot takesplace.

Step 6: Name your Watson assistant with your service.

Ex: Marketing agency service assistance.

Step 7: Enable preview link to use that to deploy it in the targeted or intended website.

Step 8: Finally click **create assistant** button.Now the skeleton of chatbot is created.

Step 9: Go to **add catalog skill** tab to create entities and intents.

Step 10: By clicking on the **create skill** tab a page will be displayed,which contains the name of created IBM Watson assistant.

Step 11: Go to the page was viewed in step 10 and add the intents and entities by giving the message templates for the intended websites .

Ex: #greetings-intent

Hello,hi,welcome,thankyou……. - entities

FEATURES:

Chatbot deployment using Watson Assistant offers several key features that enhance the conversational experience and streamline the deployment process. Here are some notable features:

Natural Language Processing (NLP): Watson Assistant leverages advanced NLP capabilities to understand and interpret user inputs, allowing the chatbot to provide accurate and contextually relevant responses.

Dialog Flow Management: With Watson Assistant, you can easily design and manage complex dialog flows. This feature enables you to create dynamic conversations, handle user intents, and guide users through multi-turn interactions.

Integration Options: Watson Assistant provides seamless integration with various channels, including websites, mobile apps, messaging platforms, and voice assistants. This flexibility allows you to deploy your chatbot across multiple platforms and reach a wider audience.

Contextual Understanding: The chatbot can maintain context throughout a conversation, remembering previous user inputs and using that information to provide more personalized and relevant responses. This feature enhances the user experience and makes interactions more natural.

Multi-language Support: Watson Assistant supports multiple languages, enabling you to deploy chatbots that can communicate with users in their preferred language. This feature is particularly useful for global deployments or multilingual user bases.

Analytics and Insights: Watson Assistant provides analytics and reporting capabilities to track user interactions, measure performance metrics, and gain insights into user behavior. This data helps you optimize the chatbot's performance and identify areas for improvement.

Continuous Learning: Watson Assistant allows you to continuously train and improve your chatbot by analyzing user interactions and incorporating user feedback. This iterative learning process helps the chatbot evolve and deliver better responses over time.

These features make Watson Assistant a powerful tool for chatbot deployment, enabling you to create intelligent and engaging conversational experiences for your users.

APPLICATIONS:

Chatbot deployment using IBM Cloud Watson Assistant can be applied in various industries and use cases. Here are some common applications:

Customer Support: Chatbots can handle customer inquiries, provide product information, troubleshoot common issues, and offer personalized recommendations. They can assist customers 24/7, reducing the need for human intervention and improving customer satisfaction.

E-commerce: Chatbots can help users navigate through product catalogs, provide recommendations based on user preferences, assist with order tracking, and answer frequently asked questions about shipping, returns, and payments.

Banking and Finance: Chatbots can handle basic banking tasks such as checking account balances, transferring funds, and providing information about banking services. They can also assist with financial planning, investment advice, and fraud detection.

Travel and Hospitality: Chatbots can assist with travel bookings, provide information about flights, hotels, and local attractions, offer personalized recommendations based on user preferences, and handle customer inquiries about reservations or cancellations.

Healthcare: Chatbots can provide basic medical information, answer health-related questions, schedule appointments, remind patients about medication, and offer guidance on common symptoms or conditions. They can also help triage patients by assessing their symptoms and suggesting appropriate actions.

Human Resources: Chatbots can assist employees with HR-related queries, such as leave requests, benefits information, policy inquiries, and onboarding processes. They can provide quick and accurate responses, freeing up HR personnel for more complex tasks.

These are just a few examples, and the applications for chatbot deployment using IBM Cloud Watson Assistant can be customized to suit specific industry needs and use cases.

BENEFITS:

There are several benefits to deploying a chatbot using IBM Cloud Watson Assistant.

Natural Language Processing (NLP): Watson Assistant utilizes advanced NLP capabilities to understand and interpret user inputs accurately. This enables the chatbot to comprehend and respond to user queries in a more human-like manner.

Easy Integration: Watson Assistant offers seamless integration with various communication channels, including websites, mobile apps, messaging platforms, and voice assistants. This allows you to deploy the chatbot across multiple platforms and reach a wider audience.

Scalability: IBM Cloud provides a scalable infrastructure, allowing your chatbot to handle a large volume of user interactions without compromising performance. You can easily scale up or down based on your needs, ensuring a smooth user experience even during peak times.

Multi-language Support: Watson Assistant supports multiple languages, enabling you to cater to a global audience. This is particularly beneficial for businesses operating in multilingual environments or targeting international markets.

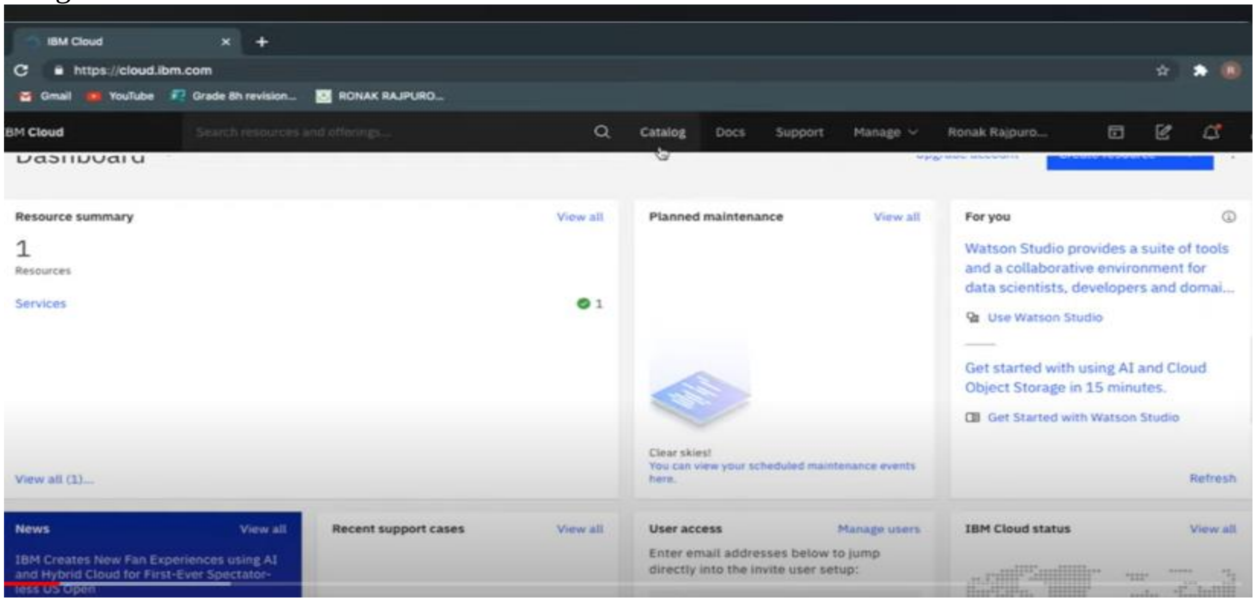
Analytics and Insights: IBM Cloud Watson Assistant provides analytics and insights into user interactions, allowing you to gain valuable insights into user behavior, preferences, and trends. This data can be used to optimize the chatbot's performance, improve user satisfaction, and make informed business decisions.

Continuous Learning: Watson Assistant allows you to continuously train and improve your chatbot by providing feedback and making updates to its knowledge base. This ensures that the chatbot stays up-to-date with the latest information and can adapt to changing user needs.

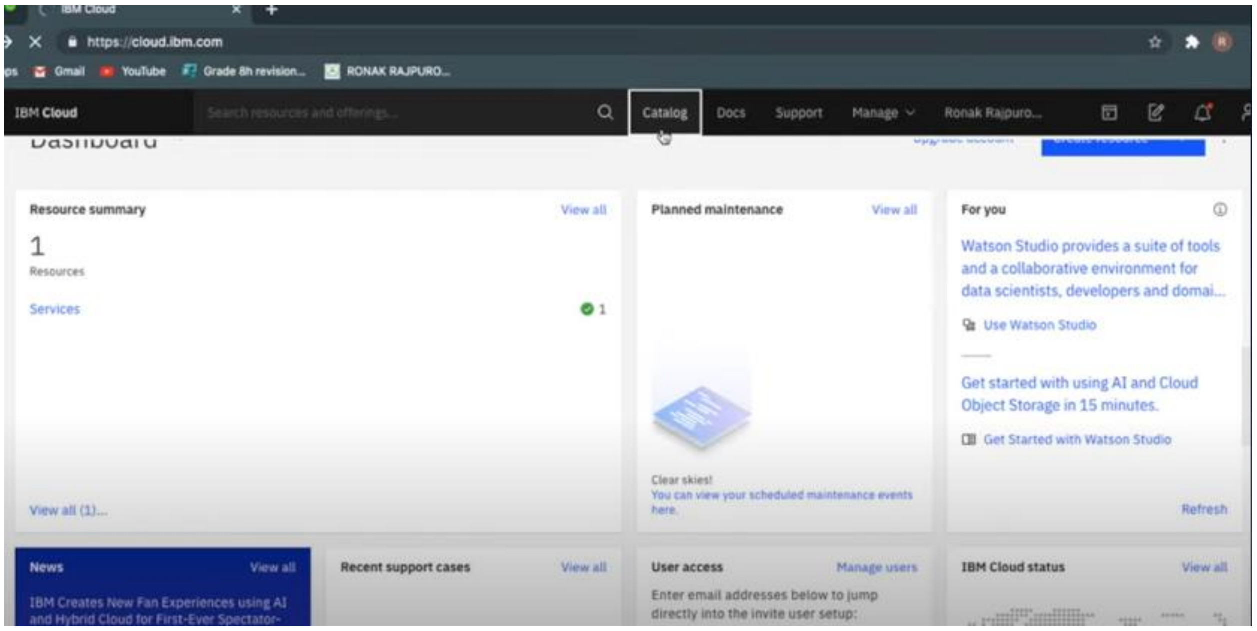
Overall, deploying a chatbot using IBM Cloud Watson Assistant offers benefits such as enhanced NLP capabilities, easy integration, scalability, multi-language support, analytics, and continuous learning, enabling you to deliver a more personalized and efficient conversational experience for your users.

SCREENSHOTS OF USER INTERFACE:

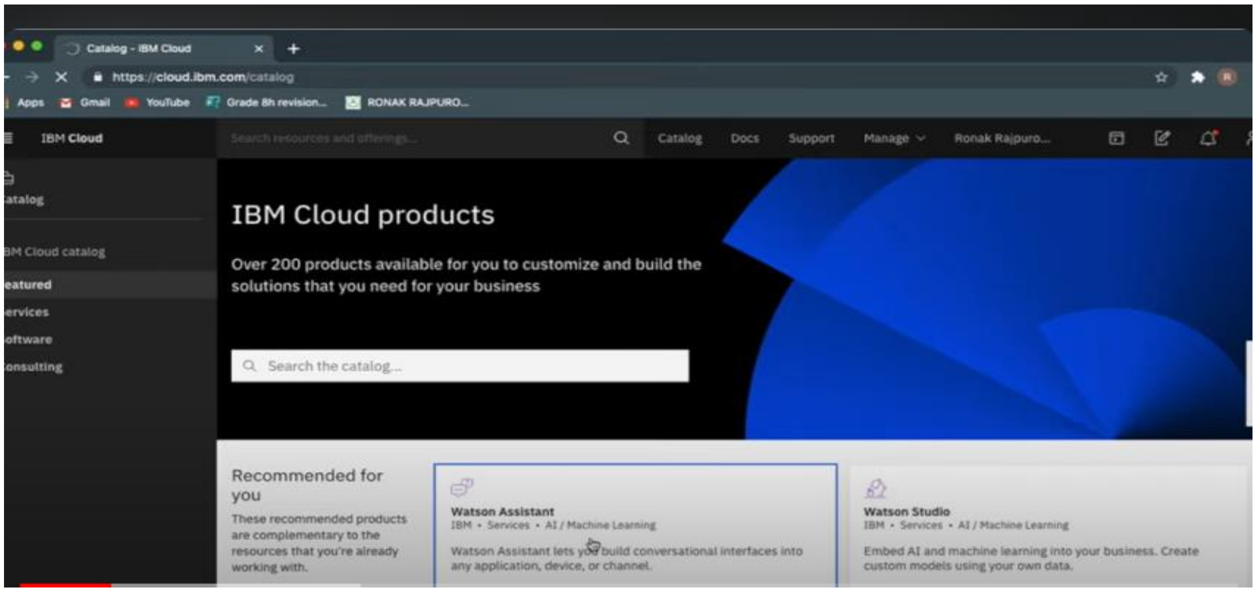
Step 1: Go t**o** IBM **cloud's** login page and create an account to access the the resources that is present in it.
If you don't have an IBM Cloud account, sign up for one or Log in to your IBM Cloud account and navigate to the IBM Watson Assistant service.


IBM Cloud
https://cloud.ibm.com
Gmail
YouTube
Grade 8h revision...
RONAK RAJPURO...
IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
Ronak Rajpuro...
Dashboard
Resource summary
View all
1
Resources
Services
View all (1)...
Planned maintenance
Clear skies!
You can view your scheduled maintenance events here.
For you
Watson Studio provides a suite of tools and a collaborative environment for data scientists, developers and domai...
Use Watson Studio
Get started with using AI and Cloud Object Storage in 15 minutes.
Get Started with Watson Studio
Refresh
News
IBM Creates New Fan Experiences using AI and Hybrid Cloud for First-Ever Spectator-less US Open
Recent support cases
User access
Manage users
Enter email addresses below to jump directly into the invite user setup:
IBM Cloud status

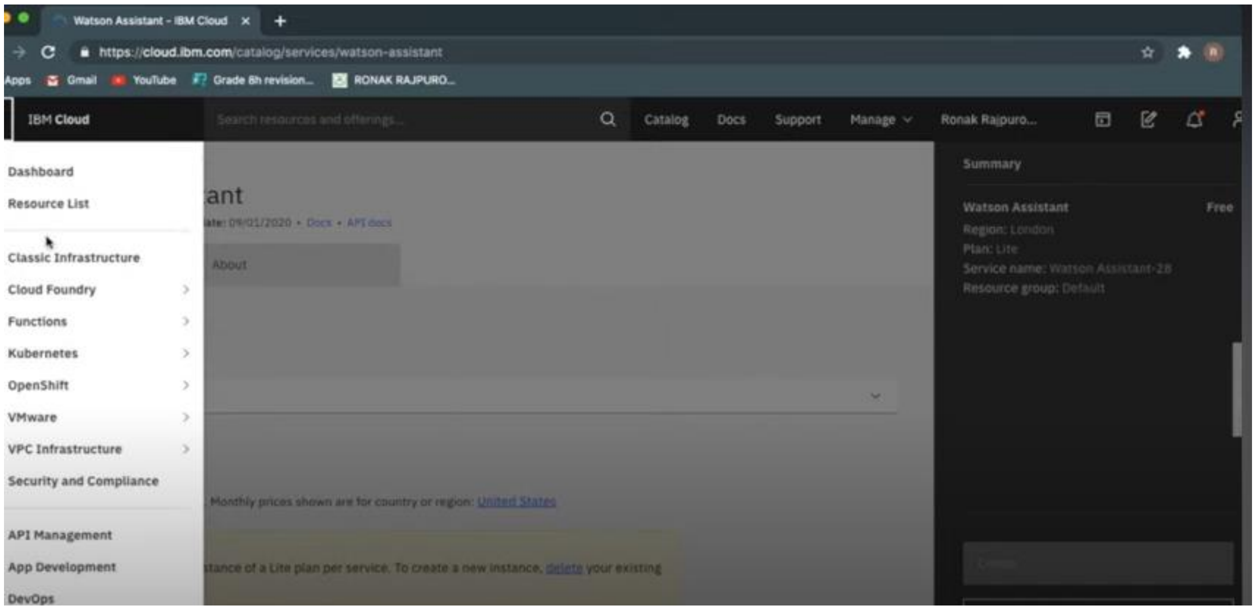
Step2: Next click on the **catalog** menu which is present on the left most corner of the web page.


IBM Cloud
Catalog
Docs
Support
Manage
Resource summary
1
Resources
Services
Planned maintenance
For you
Watson Studio provides a suite of tools and a collaborative environment for data scientists, developers and domai...
Use Watson Studio
Get started with using AI and Cloud Object Storage in 15 minutes.
Get Started with Watson Studio
Clear skies!
News
Recent support cases
User access
IBM Cloud status

Step 3: Click and move forward to **Watson assistant** tab.
Click "Create assistant."
Give your assistant a name and, optionally, a description.
Click "Create assistant."

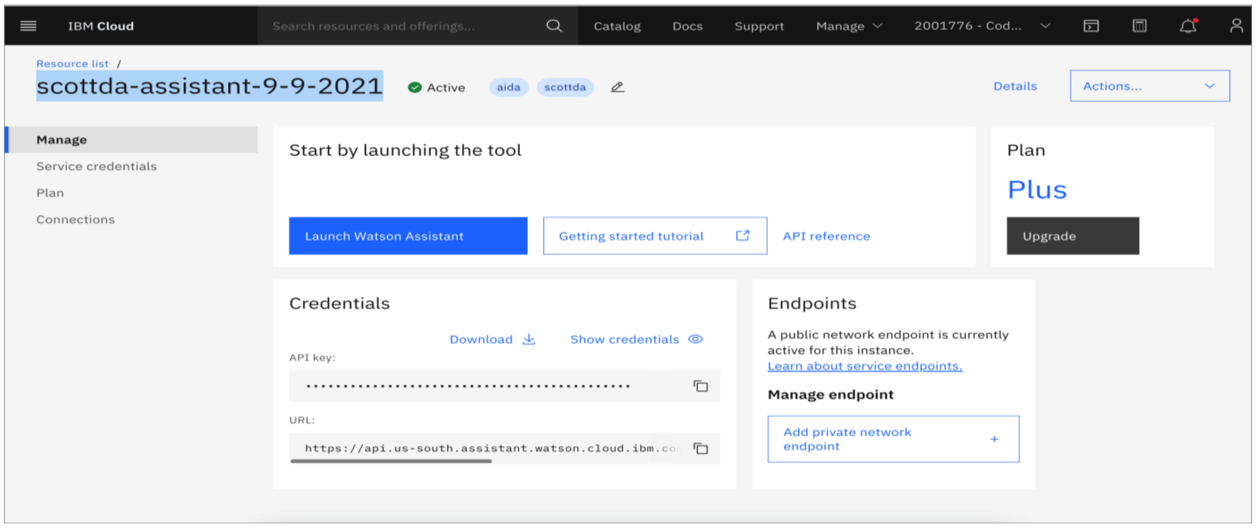

IBM Cloud products
Over 200 products available for you to customize and build the solutions that you need for your business
Search the catalog...
Recommended for you
Watson Assistant
Watson Assistant lets you build conversational interfaces into any application, device, or channel.
Watson Studio
Embed AI and machine learning into your business. Create custom models using your own data.

Step 4: Move on to **Resource list** menu and slide towards to **Services** menu item to access other services from IBM Watson assistant.



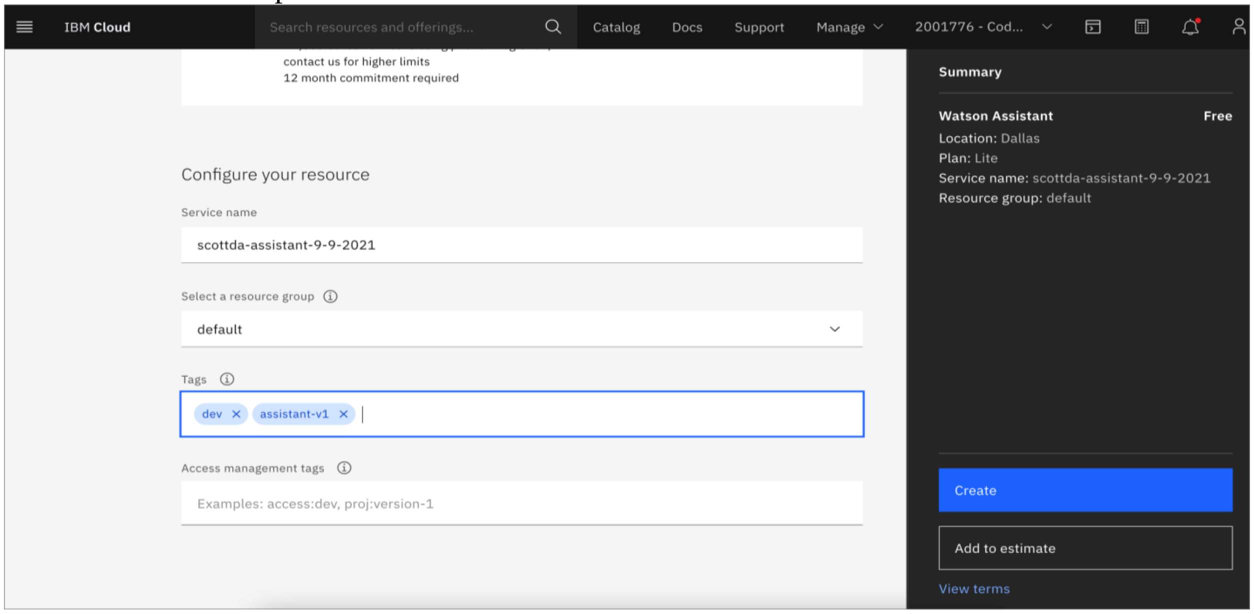
Step 5: Press **launch Watson assistant** button which will be displayed in blue.From step 5 actual deployment of chatbot takesplace.

When you launch Watson Assistant, it becomes available for use by your intended users. Launching Watson Assistant typically involves deploying the chatbot to a specific channel or platform, such as a website, messaging platform, or mobile app. Once launched, users can interact with the chatbot through the designated channel. The chatbot will use the trained models and dialog flows you have created in Watson Assistant to understand user inputs and provide appropriate responses. It can answer questions, provide information, or assist with specific tasks based on its programming and training. Launching Watson Assistant allows users to engage in conversations with the chatbot, receive automated assistance, and potentially resolve their queries or complete tasks without human intervention. It can help streamline customer support, provide information, or offer personalized recommendations, depending on how it has been configured and trained. It's important to continuously monitor and evaluate the performance of the chatbot after launching to ensure it is meeting user needs and expectations. Regular updates and improvements may be necessary based on user feedback and evolving requirements.



Step 6: Name your Watson assistant with your service.

When you name a Watson Assistant service, it is simply a way to identify and differentiate that specific instance of the Watson Assistant service within your IBM Cloud account. The name you provide does not have any direct impact on the functionality or behavior of the Watson Assistant itself. Naming the Watson Assistant service helps you easily identify and manage multiple instances of Watson Assistant if you have more than one. It can be useful when you are working with multiple projects or clients and want to keep them organized. Once you have named the Watson Assistant service, you can access and work with it through the IBM Cloud dashboard or API using the provided name. It is important to note that the actual chatbot or assistant you create within the Watson Assistant service will have its own separate name, which you can define during the assistant creation process.


IBM Cloud
Search resources and offerings...
Catalog
Docs
Support
Manage
2001776 - Cod...
contact us for higher limits
12 month commitment required
Summary
Create
Add to estimate
View terms

Configure your resource

Service name: scottda-assistant-9-9-2021

Select a resource group: default

Tags: dev, assistant-v1

Access management tags: Examples: access:dev, proj:version-1

Watson Assistant — Free
Location: Dallas
Plan: Lite
Service name: scottda-assistant-9-9-2021
Resource group: default

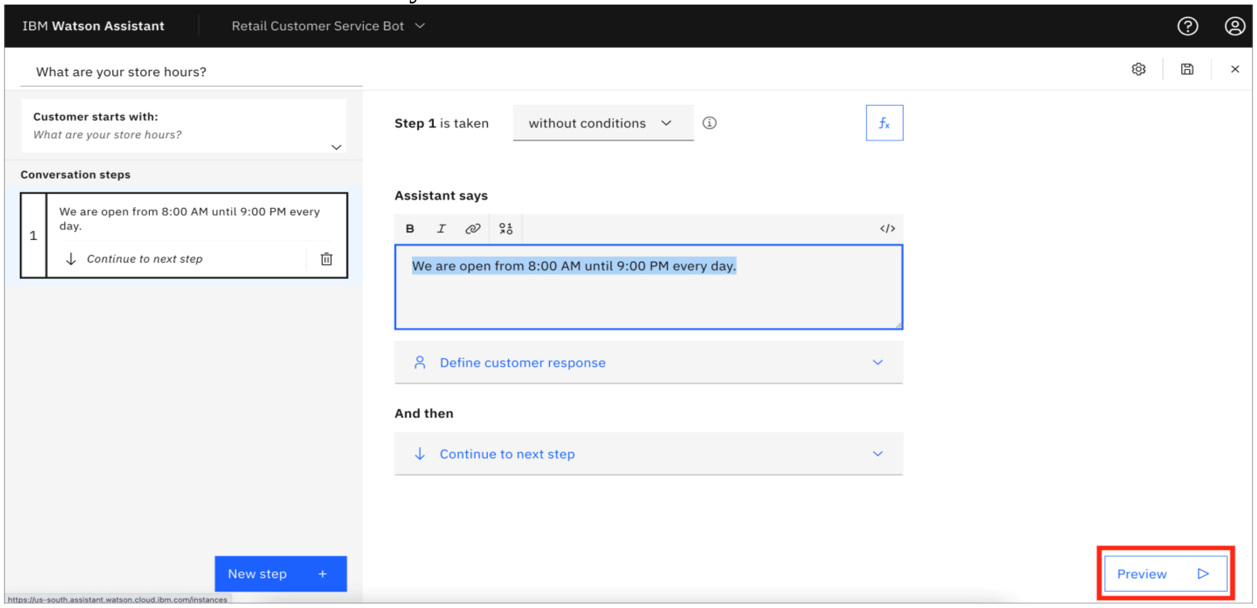
Step 7: Enable preview link to use that to deploy it in the targeted or intended website.

When you enable the preview link in Watson Assistant, it allows you to test and preview your chatbot before deploying it to production or making it available to end-users. Enabling the preview link generates a unique URL that you can use to access and interact with your chatbot in a testing environment.

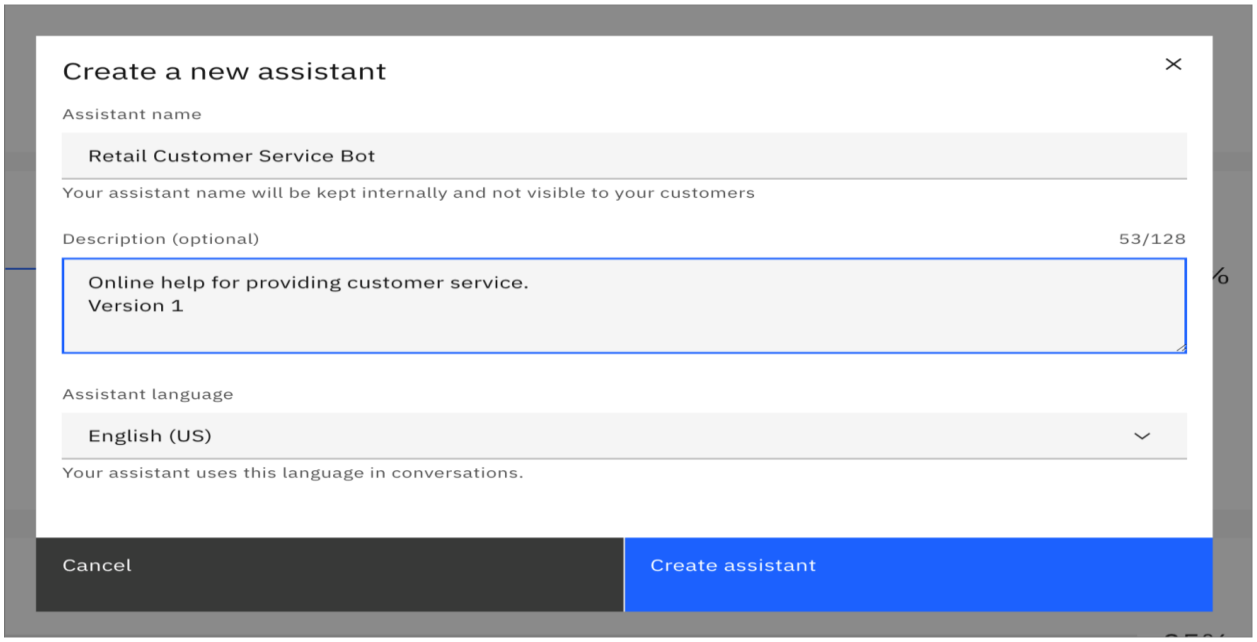
Here's what happens when you enable the preview link:

Testing environment: The preview link provides you with a dedicated testing environment where you can interact with your chatbot. This allows you to simulate conversations and test the behavior and responses of your chatbot.

Real-time changes: Any changes you make to your chatbot, such as updating dialog flows, adding intents, or modifying responses, will be reflected in the preview link. This allows you to see the impact of your changes in real-time.
Collaboration: The preview link can be shared with other team members or stakeholders, allowing them to review and provide feedback on the chatbot's functionality and behavior.
Iterative development: By using the preview link, you can iterate and refine your chatbot based on feedback and testing results. This helps ensure that your chatbot meets the desired requirements and provides a satisfactory user experience.
It's important to note that the preview link is not meant for production use or to be shared with end-users. It is solely intended for testing and preview purposes. Once you are satisfied with the performance of your chatbot, you can deploy it using the appropriate channels or integration methods to make it available to your intended audience.



Step 8: Finally click **create assistant** button.Now the skeleton of chatbot is created.
When you create an assistant in Watson Assistant, you are essentially setting up the foundation for your chatbot.
Here's a brief overview of what happens:
Name and configuration: You provide a name for your assistant and configure any additional settings, such as the language and time zone.
Skill creation: A default skill is automatically created for your assistant. A skill represents the conversational capabilities of your chatbot.
Dialog flow creation: Within the skill, you can create a dialog flow that defines how your chatbot interacts with users. This includes defining intents (user intentions), entities (relevant information), and dialog nodes (steps in the conversation).
Training and testing: You can train your assistant by providing example user inputs and mapping them to intents and entities. This helps the chatbot understand and respond accurately. You can also test your assistant to ensure it is functioning as expected.
Integration and deployment: Once your assistant is built and tested, you can integrate it with various channels like websites, messaging platforms, or mobile apps. This allows users to interact with your chatbot. You can deploy your assistant to make it available for users.
Throughout the process, you have access to the Watson Assistant tooling, which provides a user-friendly interface for creating and managing your chatbot. It also offers features like analytics and version control to help you monitor and improve the performance of your assistant over time.



Create a new assistant

Assistant name

Retail Customer Service Bot

Your assistant name will be kept internally and not visible to your customers

Description (optional) 53/128

Online help for providing customer service.
Version 1

Assistant language

English (US)

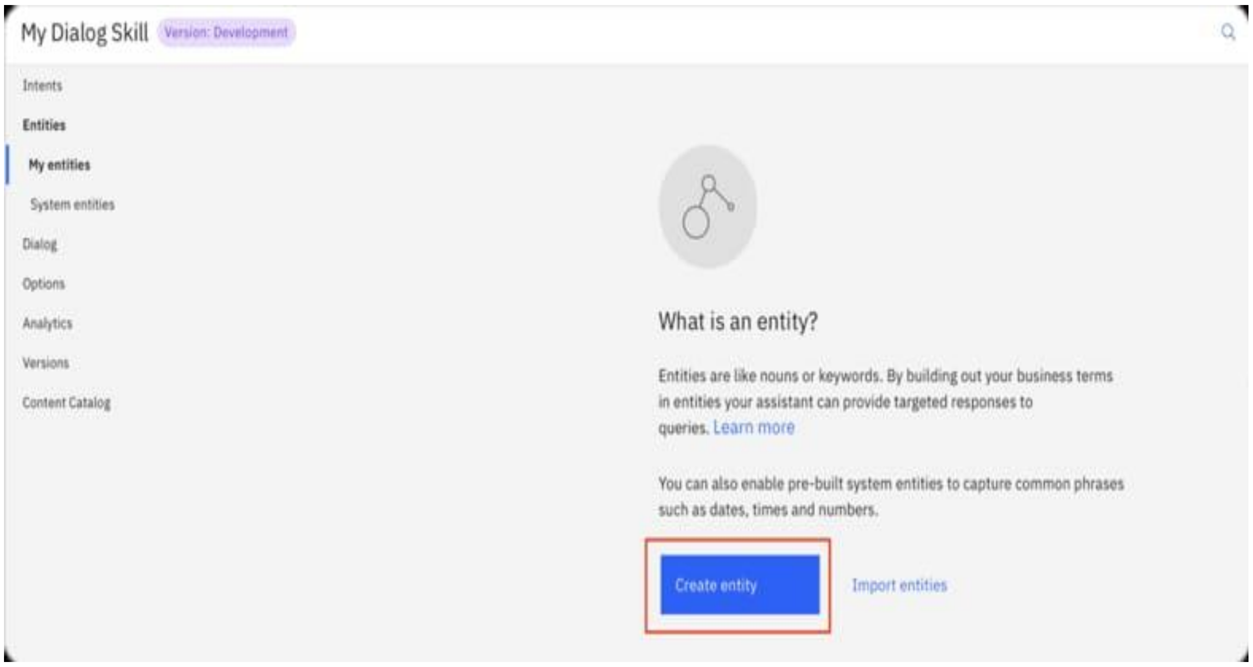
Your assistant uses this language in conversations.

Step 9: Go to **add catalog skill** tab to create entities and intents.
TO CREATE INTENTS:
In the Watson Assistant UI, click on the Skills tab.
Choose the skill you want to work with or create a new one ->Navigate to the Intents tab ->Click Create Intent. Provide a name for the intent and a brief description ->You'll then be prompted to provide example user inputs.

These are phrases or sentences users might say that relate to this intent.

TO CREATE ENTITIES:
From within the skill, navigate to the Entities tab.
Click Create Entity. Name your entity and give it a description.Next, you can add values to your entity.

For instance, if your entity is 'color', values could be 'red', 'blue', 'green', etc.


My Dialog Skill Version: Development

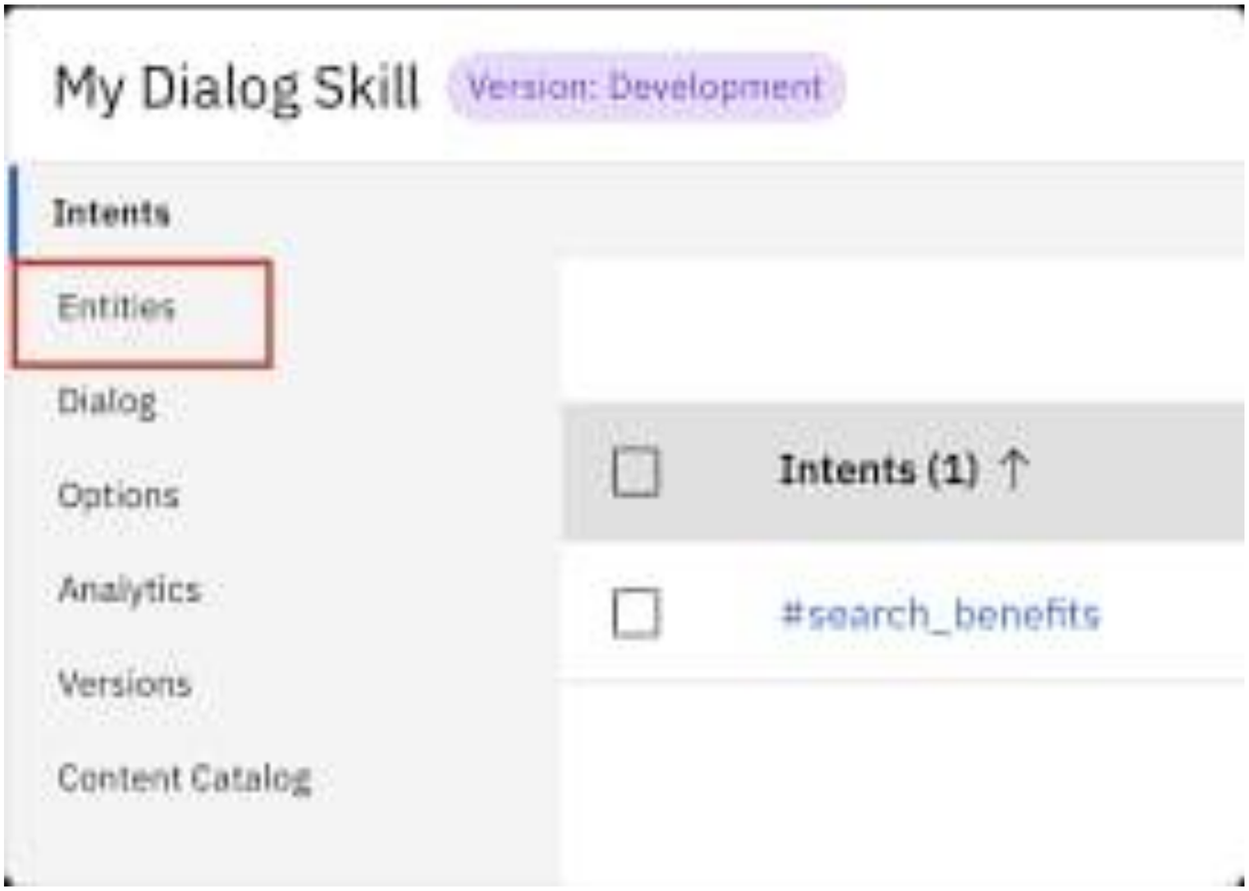
Intents
Entities
My entities
System entities
Dialog
Options
Analytics
Versions
Content Catalog

What is an entity?

Entities are like nouns or keywords. By building out your business terms in entities your assistant can provide targeted responses to queries. Learn more

You can also enable pre-built system entities to capture common phrases such as dates, times and numbers.

Create entity Import entities

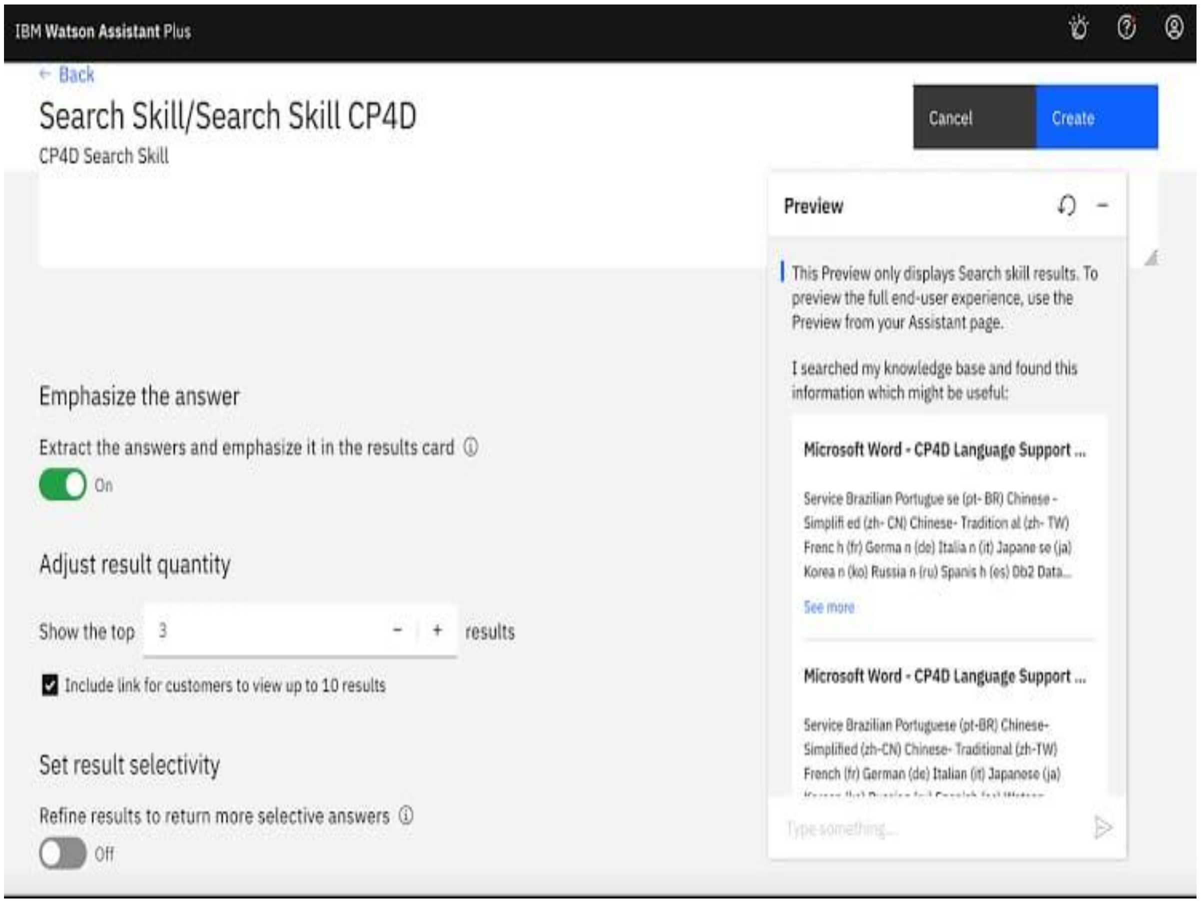

My Dialog Skill Version: Development

Intents
Entities
Dialog
Options
Analytics
Versions
Content Catalog

Intents (1) ↑
#search_benefits

Step 10: By clicking on the **create skill** tab a page will be displayed,which contains the name of created IBM Watson assistant.
SKILL NAME: This is typically a unique identifier for the specific capability or set of capabilities that the skill provides. For example, if you're building a banking chatbot, you might have skills like "AccountBalance" CONFIGURATION& DETAILS: You may see other information related to the configuration or specifics of that skill. This could include things like intents, entities, or the dialogue flow, which are critical components of how the chatbot understands and responds to user input. TRAINING DATA : Depending on the configuration, you might have the ability to upload

or input training data, which helps to improve the accuracy and effectiveness of the chatbot in recognizing and handling specific user queries TEST AND DEPLOY.: There's likely an option to test your skill in a sandbox or similar environment. Once satisfied, you can deploy the skill so that it's accessible to users.



Search Skill/Search Skill CP4D

CP4D Search Skill

Emphasize the answer

Extract the answers and emphasize it in the results card

On

Adjust result quantity

Show the top 3 results

Include link for customers to view up to 10 results

Set result selectivity

Refine results to return more selective answers

Off

This Preview only displays Search skill results. To preview the full end-user experience, use the Preview from your Assistant page.

I searched my knowledge base and found this information which might be useful:

Microsoft Word - CP4D Language Support ...

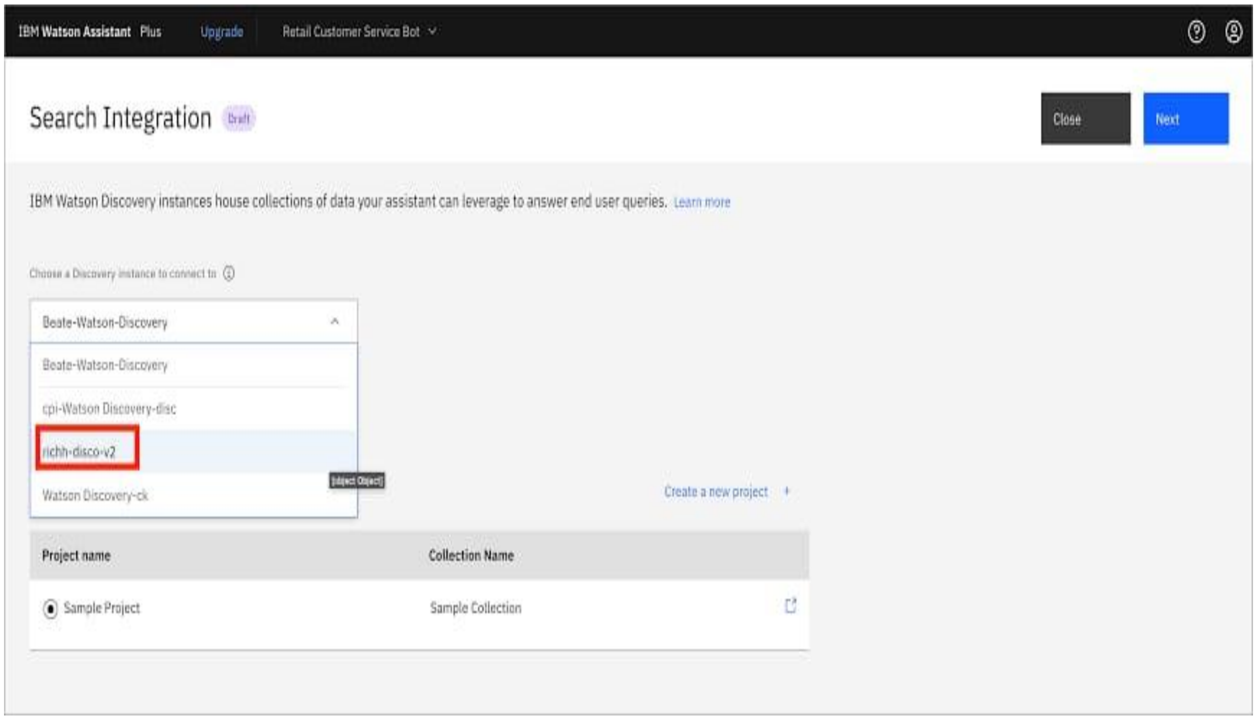
Service Brazilian Portugue se (pt- BR) Chinese - Simplifi ed (zh- CN) Chinese- Tradition al (zh- TW) Frenc h (fr) Germa n (de) Italia n (it) Japane se (ja) Korea n (ko) Russia n (ru) Spanis h (es) Db2 Data...

See more

Microsoft Word - CP4D Language Support ...

Service Brazilian Portuguese (pt-BR) Chinese- Simplified (zh-CN) Chinese- Traditional (zh-TW) French (fr) German (de) Italian (it) Japanese (ja)

Type something...



Search Integration Draft

IBM Watson Discovery instances house collections of data your assistant can leverage to answer end user queries.

Choose a Discovery instance to connect to

Beate-Watson-Discovery

Beate-Watson-Discovery

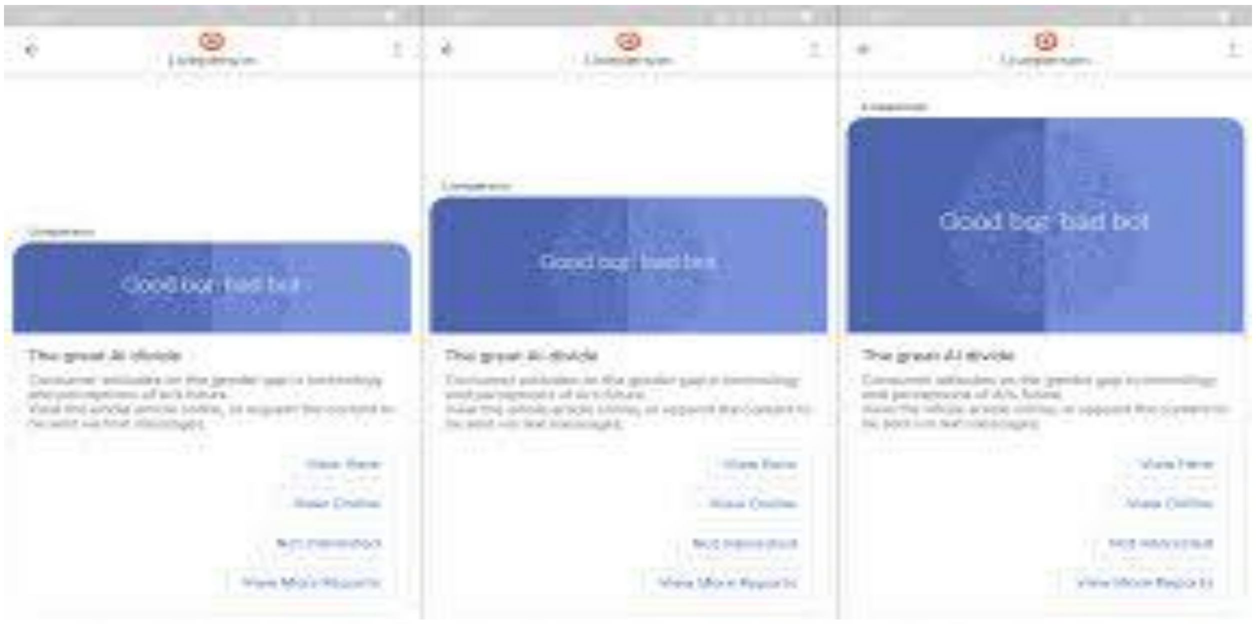
cpi-Watson Discovery-disc

richh-disco-v2

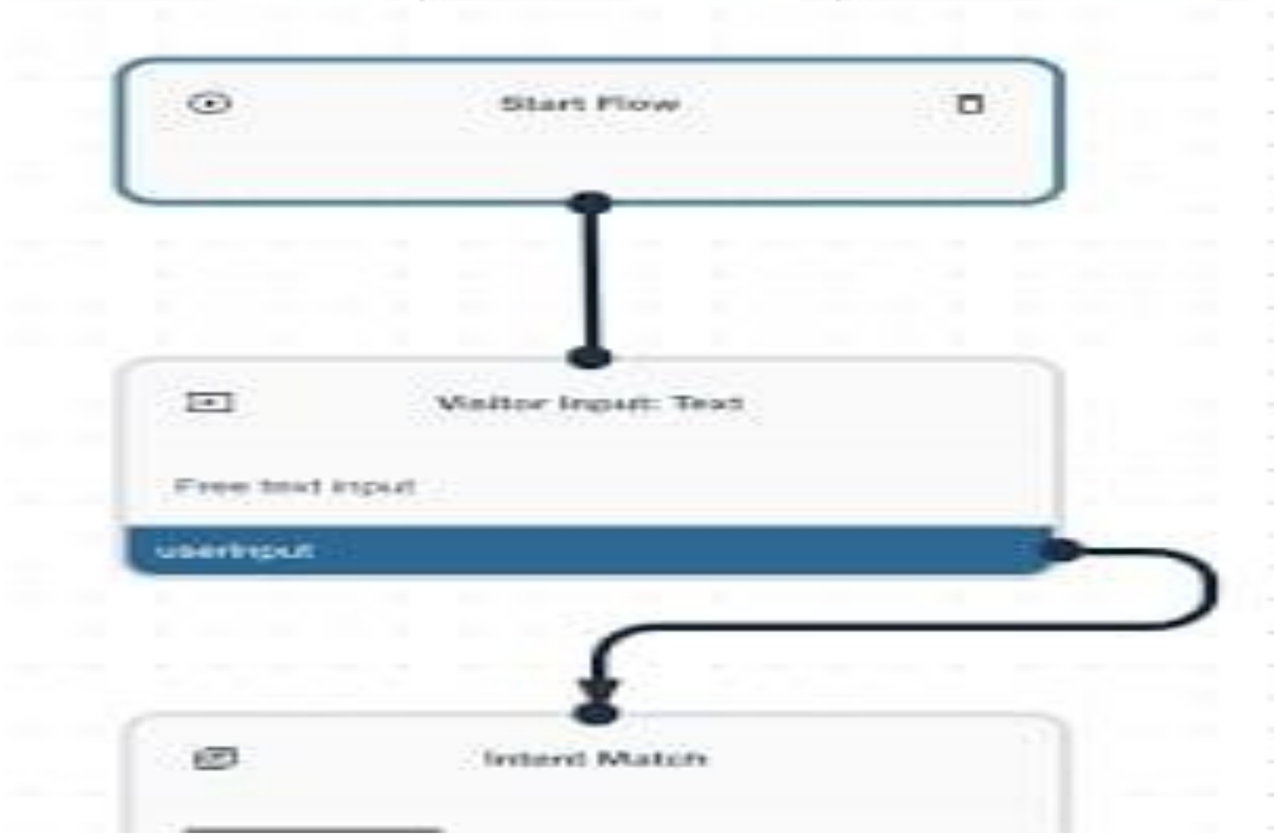
Watson Discovery-ck

Project name	Collection Name
Sample Project	Sample Collection

Step 11: Go to the page was viewed in step 10 and add the intents and entities by giving the message templates for the intended websites .



Consumer attitudes on the gender gap in technology and perceptions of AI's future.
Read the whole article online, or support the content to be sent via text messages.



FEATURE ENGINNERING:

Feature engineering is an essential step in building a chatbot. It involves transforming raw data into meaningful input features that can be used for training and improving the performance of your chatbot. Below are some feature engineering techniques for a chatbot, along with Python source code examples:

1.Text Preprocessing:

Tokenization: Split text into words or subword tokens.
Stopword Removal: Eliminate common words like "and," "the," etc.
Stemming or Lemmatization: Reduce words to their root form.

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
def preprocess_text(text):
    tokens = word_tokenize(text)
    tokens = [word for word in tokens if word.lower() not in stopwords.words('english')]
    stemmer = PorterStemmer()
    tokens = [stemmer.stem(word) for word in tokens]
    return ' '.join(tokens)
```

2.Bag of Words (BoW):
Create a BoW representation of text data.

```
from sklearn.feature_extraction.text import CountVectorizer
corpus = ["This is a sample sentence.", "Another sentence."]
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)
```

3.TF-IDF (Term Frequency-Inverse Document Frequency):
Assign weights to words based on their importance.

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus)
```

4.Word Embeddings:
Utilize pre-trained word embeddings like Word2Vec, GloVe, or FastText.

```
from gensim.models import Word2Vec
# Train Word2Vec model
model = Word2Vec(sentences, vector_size=100, window=5, min_count=1, sg=0)
```

5.Dialogue History: The chat history as context for the chatbot.

```
context = ["Hi, how can I help you?", "I'm looking for a laptop."]
user_input = "What are the best laptops?"
context.append(user_input)
```

6.Intent Classification:
pre-trained NLP model should be used for intent classification.

```
from transformers import BertTokenizer, BertForSequenceClassification
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")
model = BertForSequenceClassification.from_pretrained("bert-base-uncased")
input_text = "Tell me a joke"
inputs = tokenizer(input_text, return_tensors="pt")
outputs = model(**inputs)
```

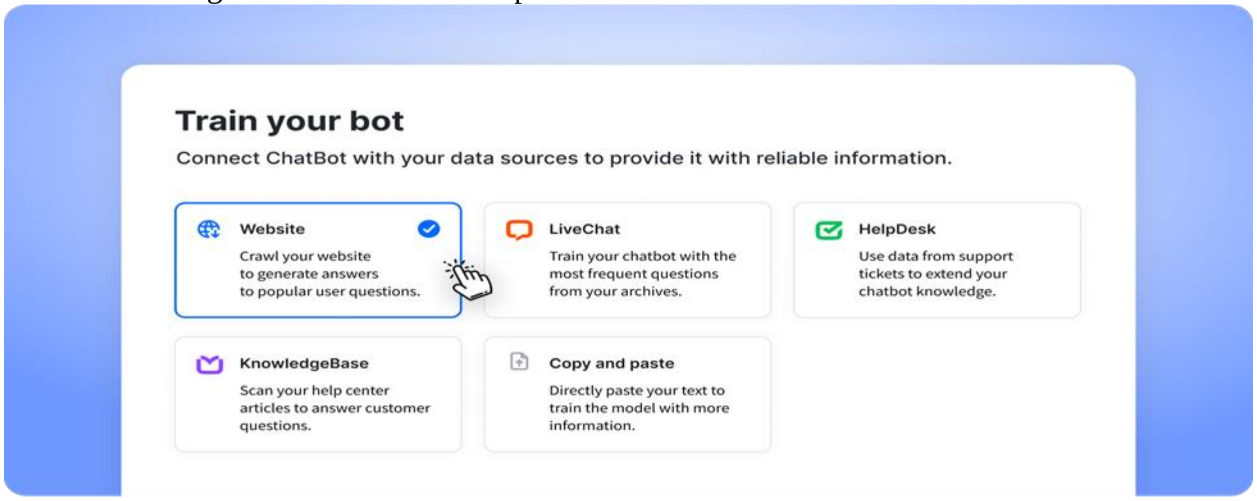
7.Named Entity Recognition (NER):

To Identify and tag entities in user messages.

```
from spacy import load
nlp = load("en_core_web_sm")
doc = nlp("I need a flight to New York on October 15th.")
entities = [(ent.text, ent.label_) for ent in doc.ents]
```

MODEL TRAINING:

Model training of a chatbot involves the process of training a machine learning model to understand and generate human-like responses in a conversational manner.



Train your bot

Connect ChatBot with your data sources to provide it with reliable information.

Website	Crawl your website to generate answers to popular user questions.
LiveChat	Train your chatbot with the most frequent questions from your archives.
HelpDesk	Use data from support tickets to extend your chatbot knowledge.
KnowledgeBase	Scan your help center articles to answer customer questions.
Copy and paste	Directly paste your text to train the model with more information.

The training typically consists of two main steps: pre-training and fine-tuning.

During pre-training, the model is exposed to a large dataset containing a wide range of text from the internet. This helps the model learn grammar, facts, and some level of reasoning. However, it is important to note that the model does not have knowledge of specific sources or the ability to verify information.

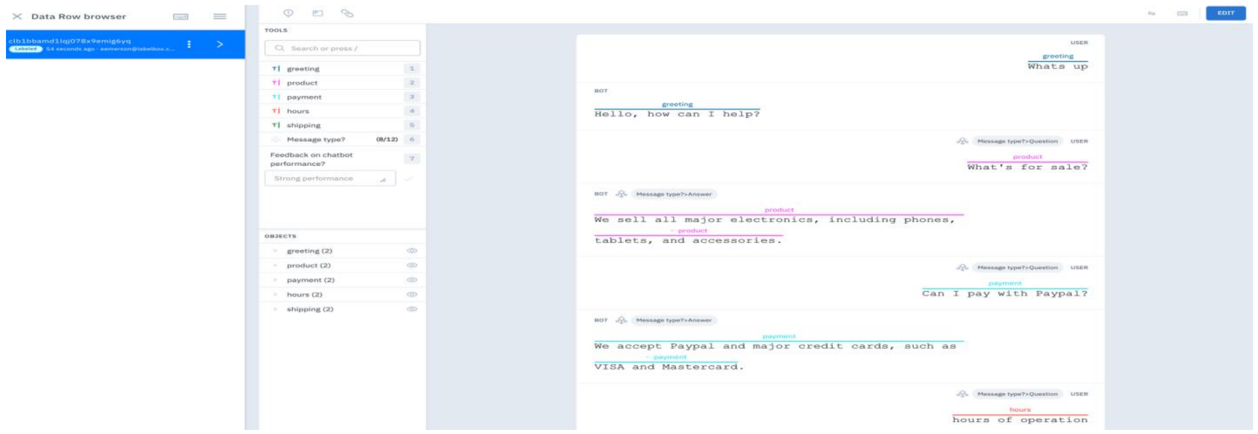
After pre-training, the model goes through fine-tuning, where it is trained on a more specific dataset that is carefully generated with the help of human reviewers. These reviewers follow guidelines provided by the developers to review and rate possible model outputs for different inputs. This iterative feedback process helps the model improve its responses over time.

It is worth mentioning that the training process aims to strike a balance between generating creative and helpful responses while avoiding biased or inappropriate content. Developers continuously work on improving the model and addressing any limitations or biases that may arise.

Overall, the model training of a chatbot involves a combination of pre-training on a large dataset and fine-tuning with human feedback to create a conversational AI that can generate human-like responses in multi-turn conversations.

Steps involved in training a chatbot

Data Collection: Gathering a large dataset of text conversations or dialogues that will be used to train the chatbot. This dataset can be obtained from various sources, such as customer support chats, online forums, or existing chat logs.



USER: Whats up

BOT: Hello, how can I help?

USER: What's for sale?

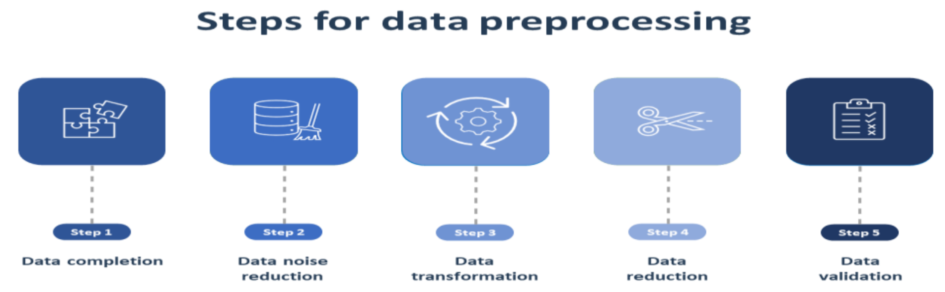
BOT: We sell all major electronics, including phones, tablets, and accessories.

USER: Can I pay with Paypal?

BOT: We accept Paypal and major credit cards, such as VISA and Mastercard.

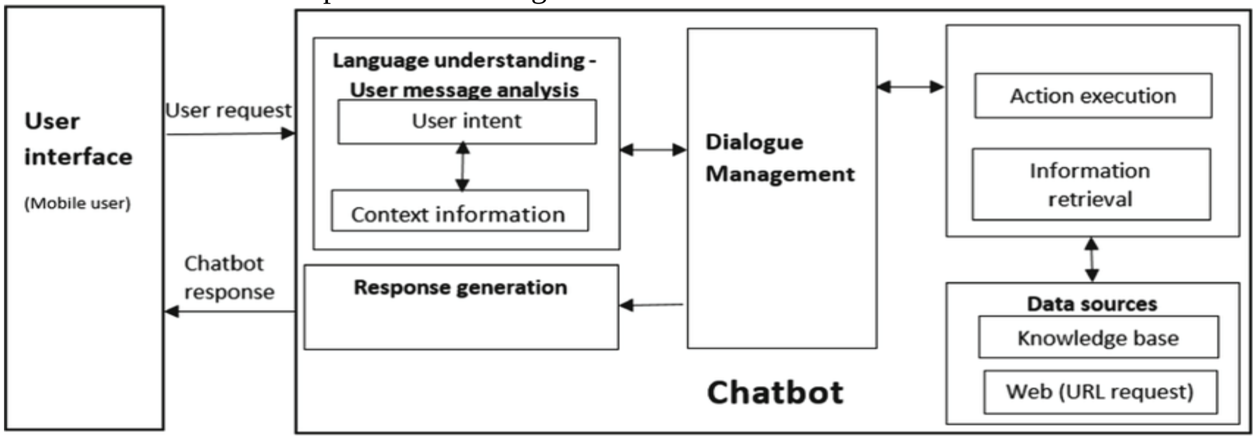
USER: hours of operation

Data Preprocessing: Cleaning and preparing the collected data for training. This step involves removing irrelevant information, formatting the data into a suitable structure, and handling any inconsistencies or noise in the dataset.



Tokenization: Breaking down the text into smaller units called tokens, such as words or subwords. Tokenization helps the model understand the structure and meaning of the text.

Model Architecture Selection: Choosing an appropriate model architecture for the chatbot. This can include recurrent neural networks (RNNs), transformer models, or a combination of both. The architecture should be capable of handling multi-turn conversations.



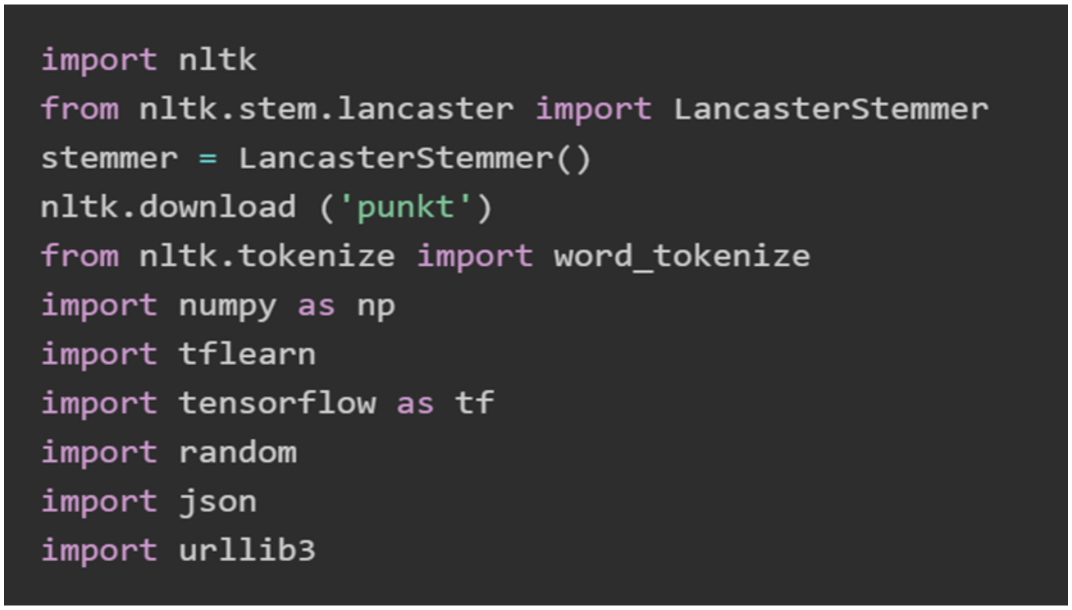
Model Training: Training the selected model using the preprocessed data. This involves feeding the input text and training the model to predict the appropriate response. The training process typically involves optimizing model parameters using techniques like backpropagation and gradient descent.

Evaluation: Assessing the performance of the trained model. This can be done by using a separate validation dataset or by conducting human evaluations to measure the quality of the generated responses.

Fine-tuning: Iteratively refining the model by incorporating feedback from human reviewers. This step helps improve the model's responses, address biases, and ensure it aligns with desired

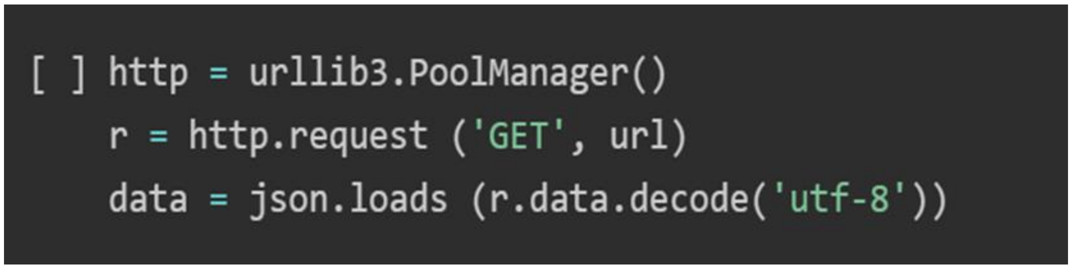
guidelines and ethical considerations.
Deployment: Integrating the trained model into a chatbot application or platform, making it available for users to interact with.
Code

- Gather and label data needed to build a chatbot
- Download and import modules



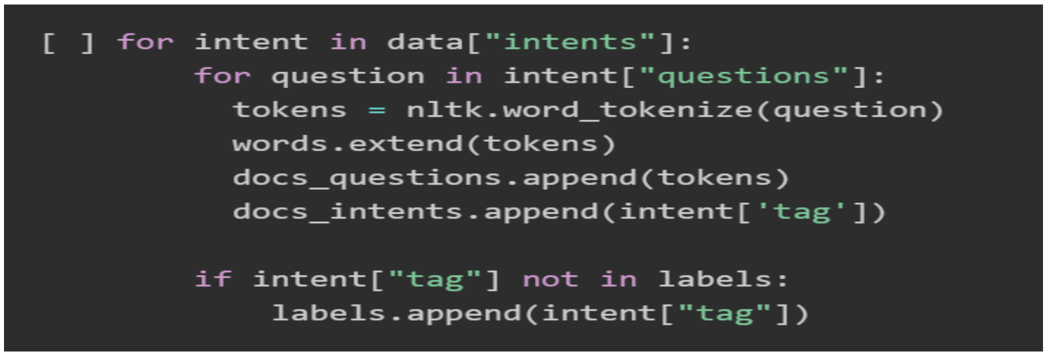
```
import nltk
from nltk.stem.lancaster import LancasterStemmer
stemmer = LancasterStemmer()
nltk.download ('punkt')
from nltk.tokenize import word_tokenize
import numpy as np
import tflearn
import tensorflow as tf
import random
import json
import urllib3
```

Modules for data



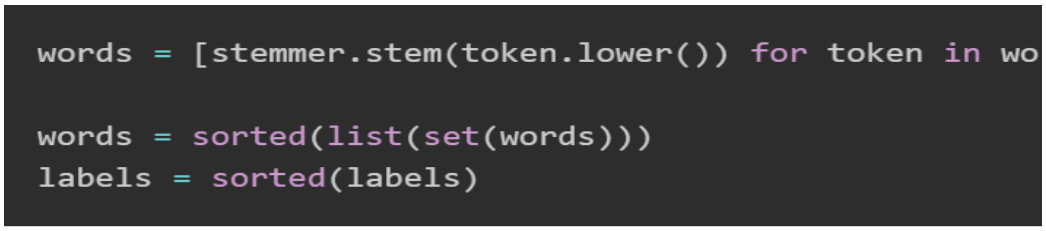
```
[ ] http = urllib3.PoolManager()
    r = http.request ('GET', url)
    data = json.loads (r.data.decode('utf-8'))
```

- Pre-processing the data
- Tockenization



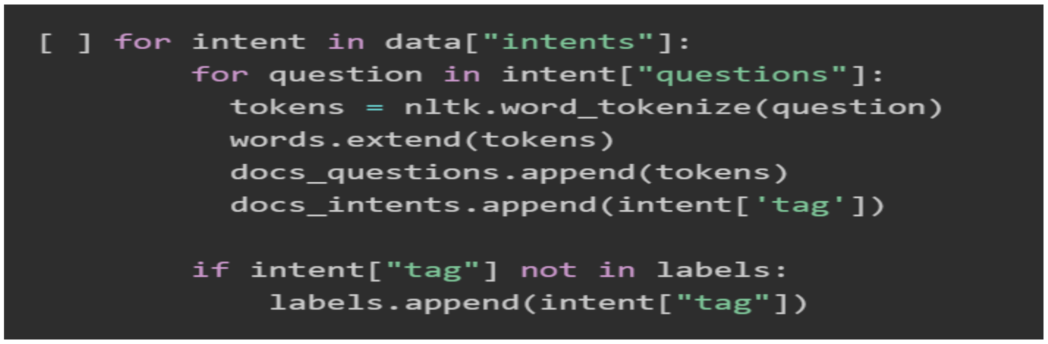
```
[ ] for intent in data["intents"]:
        for question in intent["questions"]:
          tokens = nltk.word_tokenize(question)
          words.extend(tokens)
          docs_questions.append(tokens)
          docs_intents.append(intent['tag'])

        if intent["tag"] not in labels:
            labels.append(intent["tag"])
```

- Stemming



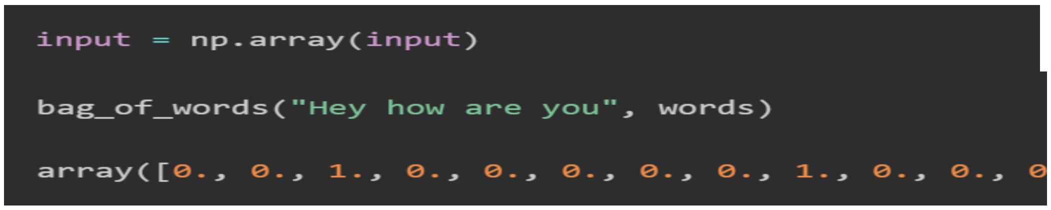
```
words = [stemmer.stem(token.lower()) for token in wo

words = sorted(list(set(words)))
labels = sorted(labels)
```

- Set up training and test the output
- Create a bag-of-words (BoW)



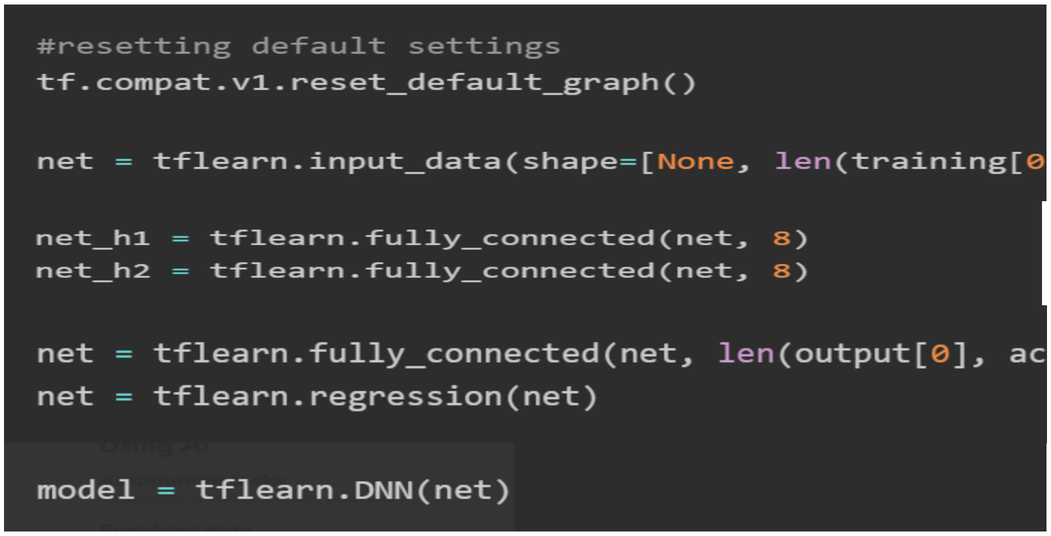
```
[ ] for intent in data["intents"]:
        for question in intent["questions"]:
          tokens = nltk.word_tokenize(question)
          words.extend(tokens)
          docs_questions.append(tokens)
          docs_intents.append(intent['tag'])

        if intent["tag"] not in labels:
            labels.append(intent["tag"])
```

- Convert BoWs into numPy arrays



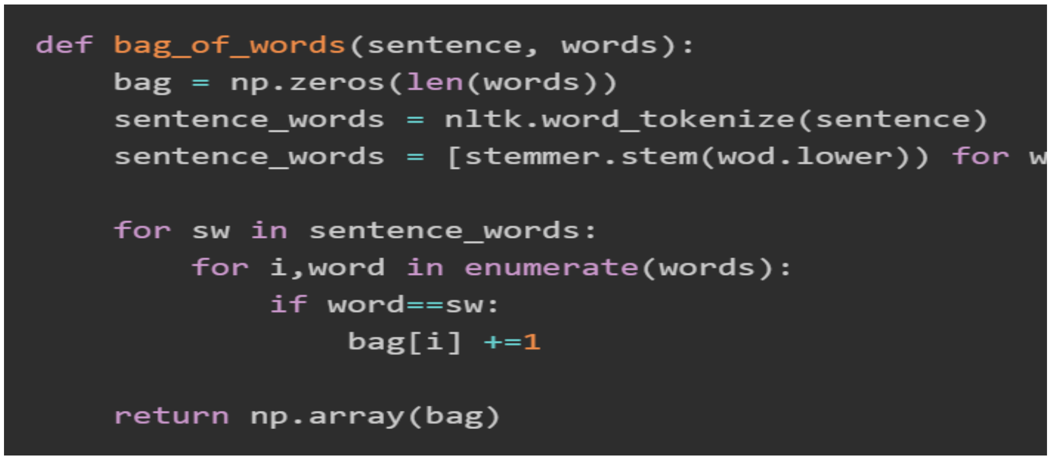
```
input = np.array(input)

bag_of_words("Hey how are you", words)

array([0., 0., 1., 0., 0., 0., 0., 0., 1., 0., 0., 0
```

- Build the model for the chatbot



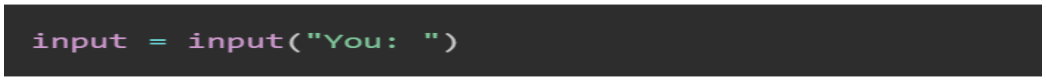
```
#resetting default settings
tf.compat.v1.reset_default_graph()

net = tflearn.input_data(shape=[None, len(training[0

net_h1 = tflearn.fully_connected(net, 8)
net_h2 = tflearn.fully_connected(net, 8)

net = tflearn.fully_connected(net, len(output[0]), ac
net = tflearn.regression(net)

model = tflearn.DNN(net)
```

- Model predictions for the chatbot



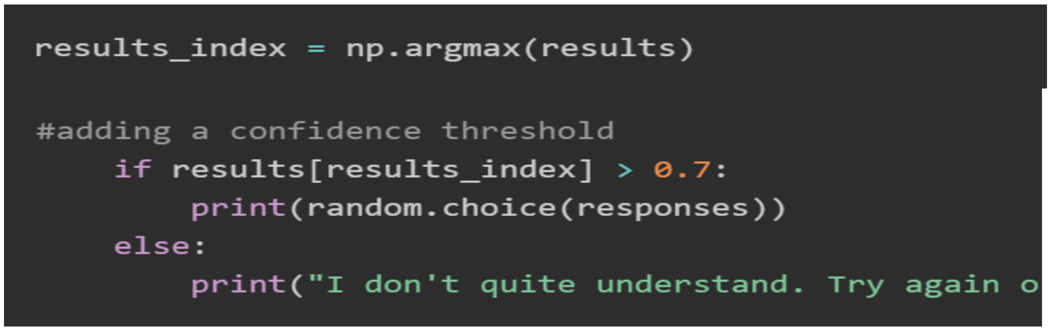
```
def bag_of_words(sentence, words):
    bag = np.zeros(len(words))
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words = [stemmer.stem(wod.lower()) for w

    for sw in sentence_words:
        for i,word in enumerate(words):
            if word==sw:
                bag[i] +=1

    return np.array(bag)
```

- Create a chat function for the chatbot



```
input = input("You: ")
```

- Classifying incoming questions for the chatbot



```
results_index = np.argmax(results)

#adding a confidence threshold
    if results[results_index] > 0.7:
        print(random.choice(responses))
    else:
        print("I don't quite understand. Try again o
```

- Customize your chatbot

EVALUATION:

Evaluating a chatbot involves assessing its performance, understanding user interactions, and measuring its effectiveness.There are several steps involved which is listed below.

1.Intent Recognition Accuracy:

Assess how accurately the chatbot recognizes user intents. Calculating the intent recognition rate using test data.

```
from sklearn.metrics import accuracy_score
    predicted_intents = chatbot.recognize_intents(test_user_inputs)
accuracy = accuracy_score(test_true_intents, predicted_intents)
```

2.Entity Recognition Accuracy:

Evaluating the chatbot's ability to correctly extract entities from user inputs.

```
from sklearn.metrics import classification_report
entity_results = chatbot.extract_entities(test_user_inputs)
print(classification_report(test_true_entities, entity_results))
```

3. User Satisfaction Surveys:

Collecting user feedback through surveys or feedback forms to understand user satisfaction and identify areas for improvement.

```
def collect_user_feedback():
   feedback = input("Please rate your experience from 1 (poor) to 5 (excellent): ")
   # Store feedback in a database or log file
```

4. Response Quality:

Chatbot should be reviewed responses for accuracy, relevance, and clarity

```
user_input = "Tell me a joke"
response = chatbot.generate_response(user_input)
print("User: ", user_input)
print("Bot: ", response)
```

5. Error Handling Assessment:

Testing the chatbot's ability to handle unexpected or unclear user inputs.

```
unclear_input = "jshsdadg&3£!?"
response = chatbot.generate_response(unclear_input)
print("User: ", unclear_input)
print("Bot: ", response)
```

6.Task Completion Rate:

Tracking the chatbot's ability is needed to help users accomplish their tasks.

```
 def check_task_completion(user_input, expected_result):
   response = chatbot.generate_response(user_input)
   if response == expected_result:
       print("Task completed successfully.")
```

CONCLUSION:

In conclusion, deploying a chatbot using IBM Cloud Watson Assistant offers a comprehensive and efficient solution for creating conversational experiences. With Watson Assistant, you can design and build your chatbot using intuitive tools and features, such as defining intents and dialog flows. The integration capabilities of Watson Assistant allow you to seamlessly deploy your chatbot across various channels, including websites, mobile apps, and messaging platforms. This ensures that your chatbot can reach and assist users wherever they are. The testing phase is crucial to ensure the functionality and accuracy of the chatbot. Watson Assistant provides robust testing capabilities, allowing you to evaluate different user scenarios and fine-tune the chatbot's responses for optimal performance. Once the chatbot is ready, deploying it to the live environment is straightforward with IBM Cloud Watson Assistant. The platform offers scalable deployment options, enabling you to handle increasing user demands and manage your chatbot effectively. Continuous monitoring and maintenance are essential to ensure the chatbot's ongoing performance and user satisfaction. Watson Assistant provides analytics and insights to track user interactions, identify areas for improvement, and make necessary adjustments to enhance the chatbot over time. Overall, deploying a chatbot using IBM Cloud Watson Assistant empowers businesses to create intelligent and effective conversational experiences, improving customer engagement and satisfaction.