INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTANDING

1-INTRODUCTION:-

In this developer code pattern, we use the typical customer care chatbot experience but instead of relying on predefined responses, our dialog will provide a hook that can call out to other IBM Watson® services for additional sources of information. In our case, it will be an owners manual that has been uploaded into Watson Discovery.

1.1 OVERVIEW:-

In this developer code pattern, we use the typical customer care chatbot experience but instead of relying on predefined responses, our dialog will provide a hook that can call out to other IBM Watson® services for additional sources of information. In our case, it will be an owners manual that has been uploaded into Watson Discovery.

In summary, this code pattern will:

- Create a customer care dialog skill in Watson Assistant
- Use Smart Document Understanding to build an enhanced Watson Discovery collection
- Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery

1.2 PURPOSE:-

The purpose of help desk is to resolve customer complaints and queries in an organized and accurate way to ensure customer satisfaction. Satisfied customers ensure a company's long-term profitability. The efficient use of help desks is also known to aid businesses in enhancing their service quality and improving the skills of service staff, especially in handling customer queries and issues.

Project Requirements:-

- *Understand your users.* Building jobs around users, take into considerations their habits and expectations.
- *Determine and document jobs.* Start with a high-level job, then identify smaller jobs which help resolve and implement a higher level job.
- *Analyze user's motivations.* Analyze what kind of benefits a user expects to get and what problem he or she wants the bot to solve in the first place. Find out how people currently solve the same problem and come up with a better way to do it.
- *Define job stories.* The key to success is to understand from the user's perspective what the entire job is and how to use a chatbot to achieve specific goals.
- *Map motivation, situations and system actions.* Considering possible situations and user motivation, you can build the flow around specific situations and provide valuable outcome.

Functional Requirements:-

- The bot should respond to any input it receives

➤ The bot doesn't understand the input, it should ask for more simplified input.

➤ If the bot understands the input, it should respond with correct information.

➤ If the bot needs more information to find an answer,and more related info

➤ The bot should use a text recognition API to understand the input to the server

- The bot should be able to send users data in the following ways:

➤ The data can be sent as a text message.

➤ The data can be sent as a graph + accompanying text, if requested by the user

- The bot should be able to query the data from the REST API.
- The bot should be able to create graphs from queried data (if the graph is deemed helpful).

INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTANDING

Technical Requirements:-

1. The user interacts with the back-end server via the app UI. The front-end app UI is a chatbot that engages the user in a conversation.
2. Dialog between the user and back-end server is coordinated using a Watson Assistant dialog skill.
3. If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functions action.
4. The IBM Cloud Functions action will query the Watson Discovery Service and return the results.

Software Requirements:-

- Artificial Intelligence
- Node.js
- Serverless
- Webhooks
- Node red application
- Watson Assistant
- Watson Discovery
- IBM cloud

2 - LITERATURE SURVEY:-

2.1 EXISTING PROBLEM:-

- The typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn't valid or offer to speak to a real person.

- The Existing problem with the customer help desk that is the when the customer asked the query which is not a direct message.They need information about the particular product and i.e(how to turn on the heater ,maximum temperature of heater).

- This is not able to answer the query by the human bot.So,for the customer its shows a irrelevant query

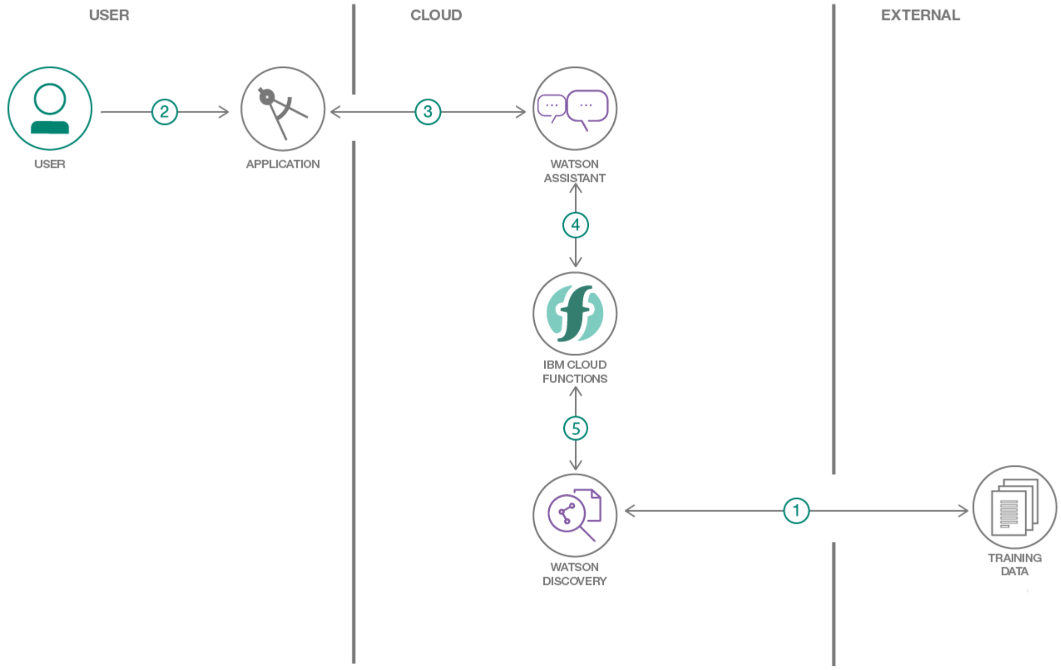
2.1 PROPOSED SOLUTION:-

- In our code pattern, we will provide another option. If the customer question is about the operation of a device, we will use the webhook feature of Watson Assistant to pass the question onto our Watson Discovery Service, which has been pre-loaded with the device's owners manual. So now, instead of "Would you like to speak to a customer representative?" we can return relevant sections of the owners manual to help solve our customers' problems.
- To take it a step further, we will use the Smart Document Understanding feature of Watson Discovery to train it on what text in the owners manual is important and what is not. This will improve the answers returned from the queries.

INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTANDING

3 -THEORITICAL ANALYSIS:-

3.1 BLOCK DIAGRAM:-



3.2 HARDWARE/SOFTWARE DESIGNING:-

In this project,Intelligent Customer help desk using smart document inderstanding we use several types of software.

- Login:creating an IBM cloud Account
- Create a Node-RED Starter application running in the IBM Cloud Secure the application
- Exploring the watson usecases
- Watson Assisstant:Watson Assistant is a conversation AI platform that helps you provide customers fast, straightforward and accurate answers to their questions, across any application, device or channel

- Watson Discovery:Using the Watson Discovery Smart Document Understanding (SDU) feature, we will enhance the Discovery model so that queries will be better focused to only search the most relevant information found in a typical owner's manual.SDU:SDU trains Watson Discovery to extract custom fields in your documents. Customizing how your documents are indexed into Discovery will improve the answers returned from queries.
  With SDU, you annotate fields within your documents to train custom conversion models. As you annotate, Watson is learning and will start predicting annotations. SDU models can also be exported and used on other collections.

Steps:

1. Clone the repo
2. Create IBM Cloud services
3. Configure Watson Discovery
4. Create IBM Cloud Functions action
5. Configure Watson Assistant
6. Get IBM Cloud services credentials and add to .env file
7. Run the application

INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTANDING

4-EXPERIMENTAL INVESTIGATION:-

Import the document

As shown below, launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option. Give the data collection a unique name. When prompted, select and upload the ecobee3_UserGuide.pdf file located in the data directory of your local repo.

The Ecobee is a popular residential thermostat that has a wifi interface and multiple configuration options.

Before applying SDU to our document, lets do some simple queries on the data so that we can compare it to results found after applying SDU.

Enter queries related to the operation of the thermostat and view the results. As you will see, the results are not very useful, and in some cases, not even related to the question.

Now let's apply SDU to our document to see if we can generate some better query responses.

From the Discovery collection panel, click the Configure data button (located in the top right corner) to start the SDU process.

Here is the layout of the Identify fields tab of the SDU annotation panel:

The goal is to annotate all of the pages in the document so Discovery can learn what text is important, and what text can be ignored.

Once you click the Apply changes to collection button [6], you will be asked to reload the document. Choose the same owner's manual .pdf document as before.

Store credentials for future use

In upcoming steps, you will need to provide the credentials to access your Discovery collection. The values can be found in the following locations.

The Collection ID and Environment ID values can be found by clicking the dropdown button [1] located at the top right side of your collection panel

Create IBM Cloud Functions action

Now let's create the web action that will make queries against our Discovery collection.

Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. Enter functions as the filter, then select the Functions card

From the Functions main panel, click on the Actions tab. Then click on Create.

From the Create panel, select the Create Action option.

On the Create Action panel, provide a unique Action Name [1], keep the default package [2], and select the Node.js 10 [3] runtime. Click the Create button [4] to create the action.

Add the following keys:

- url
- environment_id
- collection_id
- iam_apikey

This dialog skill contains all of the nodes needed to have a typical call center conversation with a user.

Add new intent

The default customer care dialog does not have a way to deal with any questions involving outside resources, so we will need to add this.

Create a new intent that can detect when the user is asking about operating the Ecobee thermostat.

From the Customer Care Sample Skill panel, select the Intents tab.

Click the Create intent button.

INTELLIGENT CUSTOMER HELP DESK WITH SMART DOCUMENT UNDERSTANDING

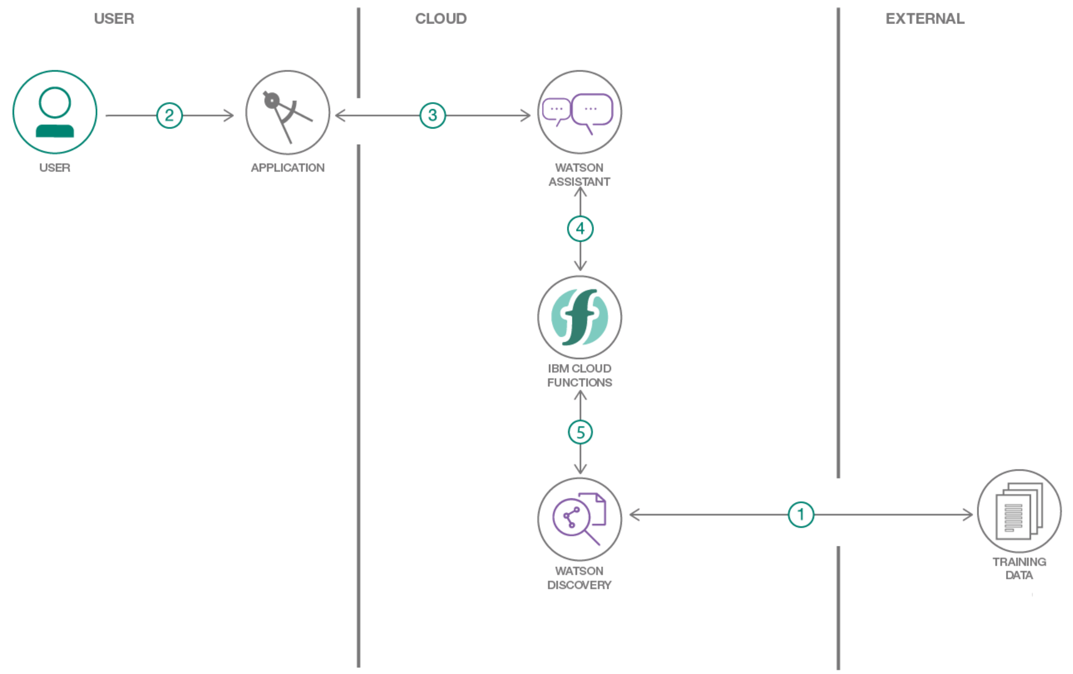
Credentials can be found by clicking the Service Credentials tab, then the View Credentials option from the panel of your created Watson service.

An additional ASSISTANT_SKILL_ID value is required to access the Watson Assistant service. To get this value, select the Manage tab, then the Launch tool button from the panel of your Watson Assistance service. From the service instance panel, select your Assistant to display the assigned skills. For this code pattern, we used the dialog skill named Custom Skill Sample Skill that comes with the service:

Create the workflow in node-red application and Run the application

Test the bot and capture the result

FLOWCHART:-



RESULT:-

Here finally the output for customer help desk with smatr document understanding was displayed.Here bot answer to the customer queries.

ADVANTAGES:-

- In this customer chatbot able to know the very basic queries and also about the product information in a simple step by step procedure.
- Any number of queries they can be able to ask I they got the immediate response

DISADVANTAGES:-

- The disadvantage with the chatbot is if the query sked by the customer excluding basic information and product information the bot not able to answer the query
- The chatbot is quite simples

APPLICATIONS:-

- Customer help center
- Market research
- product services

CONCLUSIONS:-

- Hereby,the by using this customer help desk the customer able to ask the queries whenever because it's a 24 x 7 customer service

FUTURE SCOPE:-

- The future scope of this project is to make bot well trained and able to answer each and every queries asked by the customer.Improve the design of the chatbot for making convinient to the available customers
- And taking this project to be applicable to the some other application services like ticket booking,e-commerce etc