

1.INTRODUCTION

1.1 Overview :

We will be able to write an application that leverages multiple Watson AI Services (Discovery, Assistant, Cloud function and Node Red). By the end of the project, we'll learn best practices of combining Watson services, and how they can build interactive information retrieval systems with Discovery + Assistant.

1. Project Requirements: Python, IBM Cloud, IBM Watson, Node- RED
2. Functional Requirements: IBM cloud
3. Technical Requirements: AI, ML, WATSON AI, PYTHON
4. Software Requirements: Watson assistant, Watson discovery.
5. Project Deliverables: Smartinternz Internship
6. Project Duration: 30 days

1.2 Purpose:

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

1.2.1 Scope of Work:

1. Create a customer care dialog skill in Watson Assistant
2. Use Smart Document Understanding to build an enhanced Watson Discovery collection
3. Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to Watson Discovery
4. Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

2. LITERATURE SURVEY

2.1 Existing problem:

Generally Chatbots means getting input from users and getting only response questions and for some questions the output from bot will be like “try again”, “I don’t understand”, “will you repeat again”, and so on... and directs customer to customer agent but a good customer Chatbot should minimize involvement of customer agent to chat with customer to clarify his/her doubts. So to achieve this we should include an virtual agent in chatbot so that it will take care of real involvement of customer agent and customer can clarify his doubts with fast chatbots.

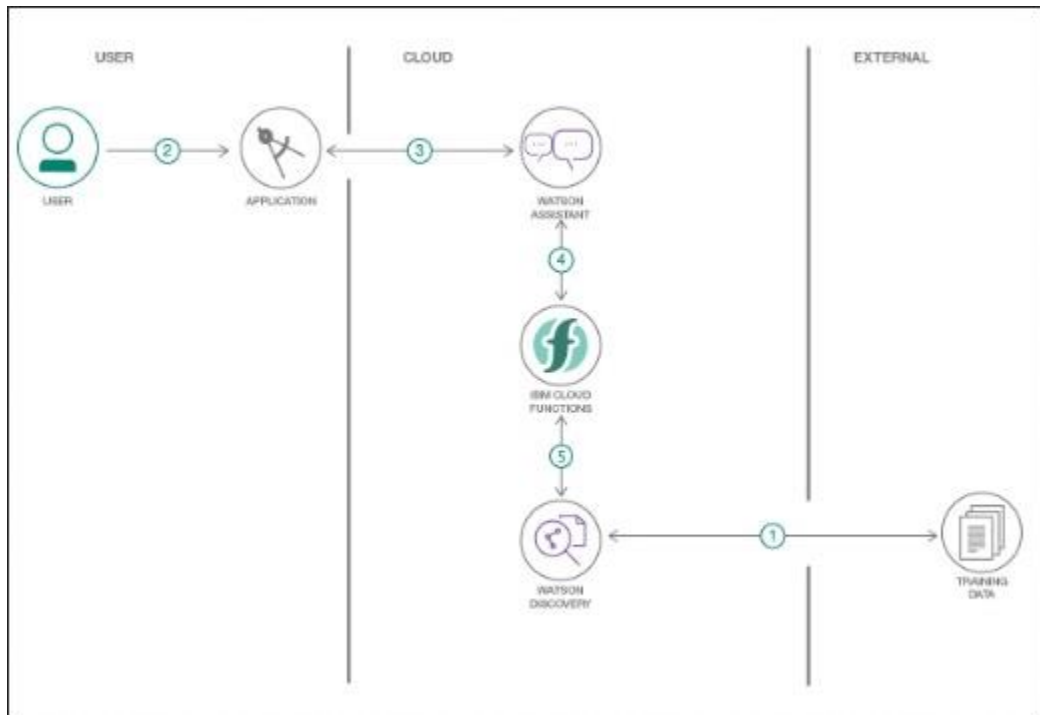
2.2 Proposed solution:

For the above problem to get solved we have to put an virtual agent in chatbot so it can understand the queries that are posted by customers. The virtual agent should be trained from some insight records based company background so it can answer queries based on the product or related to company. In this project I used WatsonDiscovery to achieve the above solution. And later including Assistant and Discovery on Node-RED

3. THEORETICAL ANALYSIS

3.1 Block/Flow Diagram:

The following flow is the basic working flow of the project.



3.2 Software designing:

1. Create IBM Cloud services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action
4. Configure Watson Assistant
5. Create flow and configure node
6. Deploy and run Node Red app.

4. EXPERIMENTAL PROCEDURE

1. Create IBM Cloud services.
- Create the following services:

- Watson Discovery
- Watson Assistant
- Node Red

2. Configure Watson Discovery

1. Import the document

Launch the Watson Discovery tool and create a new data collection by selecting the Upload your own data option.

2. Give the data collection a unique name.

3. When prompted, select and upload the Ecobee3 user guide in pdf format.

The Ecobee3 is a popular residential thermostat that has a Wi-Fi interface and multiple configuration options.

4. Before applying SDU to our document, let's 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.

4. Annotate with SDU. 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.

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

The screenshot shows the IBM Watson Discovery interface for the 'Identify fields' tab. The top navigation bar includes 'Overview', 'Errors and warnings (129)', and 'Search settings'. The main header displays '129 documents', '0 documents failed', and creation/update timestamps. A sidebar on the left lists identified fields: footer, subtitle, table_of_contents, text, and title. The main content area is divided into three sections: 'Identified 5 fields from your data', 'Added 4 enrichments to your data' (showing Entity Extraction results like '0.3°C (4)', '0.5°F (4)', '10 °F (4)', '900 seconds (4)', '20 min (3)') and 'Sentiment Analysis' results (55% positive, 31% neutral, 13% negative). On the right, a 'Now you're ready to query!' section offers three query options: 'Top entities with their average, min, max sentiment score', 'Most common entity types and their top entities', and 'Documents that contain HVAC, but not Heat'. Each option has a 'Run' button.

The goal is to annotate all the pages in the document so that Discovery can learn what text is important, and what text can be ignored. Follow the following instructions to achieve this task:

- [1] is the list of pages in the manual. As each is processed, a green check mark will appear on the page.
- [2] is the current page being annotated.
- [3] is where you select text and assign it a label.
- [4] is the list of labels you can assign to the page text.
- Click [5] to submit the page to Discovery.
- Click [6] when you have completed the annotation process.

3. Create IBM Cloud Functions action

Now let us 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 [1], then select the Functions card [2]

- 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.
- Once your action is created, click on the Code tab [1]:
 - In the code editor window [2], cut and paste in the code from the cloud_function.js file found in the actions directory of your local repository. The code is pretty straightforward - it simply connects to the Discovery service, makes a query against the collection, then returns the response.

5.FLOWCHART

At first, go to manage palette and install dashboard.

Now, Create the flow with the help of following node:

- Template
- Assistant

- Debug
- Function
- Ui_Form

6. RESULTS

Finally, our Node-RED dash board integrates all the components and displayed in the Dashboard UI by typing URL.

7. ADVANTAGES & DISADVANTAGES

Advantages:

1. Companies can deploy chatbots to rectify simple and general human queries.
2. Reduces man power.
3. Cost efficient.
4. No need to divert calls to customer agent and customer agent can look on other works.

Disadvantages:

1. Sometimes chat bot can mislead customers.
2. Giving same answer for different sentiments.
3. Sometimes cannot connect to customer sentiments and intentions.
4. It can be deployed on many popular social media applications like Facebook, Slack, Telegram.
5. The Chatbot can deploy any website to clarify basic doubts of viewers

9. CONCLUSION

Thus, we have successfully created the Intelligent Customer Help Desk Smart Chatbot using Watson Assistant, Watson Discovery Services, Node-RED and Cloud-functions.

10. FUTURE SCOPE

We can include Watson Studio Text to Speech and Speech to Text services to access the chatbot hands free. This is one of the future scope of this project.