

## **Project Report**

**Name:** Nishi Gothi (nishigothi@gmail.com)

**Title:** Intelligent Customer Help Desk with Smart  
Document Understanding

**Category:** Artificial Intelligence

**Company Name:** SmartInternz

**Year:** 2020

# Introduction

## Project Overview:

The typical customer care chat bot 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 predetermined 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 preloaded 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 customer's 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.

**Project Requirements:** Python, IBM Cloud, IBMWatson

**Functional Requirements:** IBMcloud

**Technical Requirements:** AI, ML, WATSONAI, PYTHON

**Software Requirements:** Watson assistant, Watsondiscovery.

## Scope of Work

- a. Create a customer care dialog skill in WatsonAssistant
- b. Use Smart Document Understanding to build an enhanced Watson Discovery collection
- c. Create an IBM Cloud Functions web action that allows Watson Assistant to post queries to WatsonDiscovery.
- d. Build a web application with integration to all these services & deploy the same on IBM CloudPlatform

# Literature Survey

## a. Proposed Problem

The typical customer care chat bot 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 predetermined question set, the option is typically to tell the customer the question isn't valid or offer to speak to a real person.

## b. Proposed Solution

### Steps

- a. Create IBM Cloudservices
- b. Configure WatsonDiscovery
- c. Create IBM Cloud Functionsaction.
- d. Create a Node red flow to connect all the services together.
- e. Configure WatsonAssistant.
- f. Create flow and configure node
- g. Deploy and run Node Redapp.

### 1. Create IBM Cloudservices

Create the following services:

- a. WatsonDiscovery
- b. WatsonAssistant
- c. NodeRed

## 2. Configure WatsonDiscovery

### *Import the document*

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 redmi\_note\_5\_pro\_manual.pdf file located in the data directory of your local repo.

### *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. 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.

## 3. Create IBM Cloud Functionsaction

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 Clouddashboard. 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, keep the default package and select the Node.js 10 runtime.

Click the Create button to create the action.

Once your action is created, click on the Code tab:

In the code editor window, cut and paste in the code from the disco-action.js file found in the action's directory of your local repository. The code is pretty straight-forward - it simply

connects to the Discovery service, makes a query against the collection, then returns the response.

If you press the Invoke button, it will fail due to credentials not being defined yet. We'll do this next. Select the Parameters tab:

Add the following keys:

url

environment\_id

collection\_id

iam\_apikey

For values, please use the values associated with the Discovery service you created in the previous step. Now that the credentials are set, return to the Code panel and press the Invoke button again. Now you should see actual results returned from the Discovery service:

Next, go to the Endpoints panel:

Click the checkbox for Enable as Web Action. This will generate a public endpoint URL.

Take note of the URL value, as this will be needed by Watson Assistant in a future step.

To verify you have entered the correct Discovery parameters, execute the provided curl command. If it fails, re-check your parameter values.

#### **4. Configure WatsonAssistant**

Launch the Watson Assistant tool and create a new dialog skill. Select the Use sample skill option as your starting point. 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 Product. From the Customer Care Sample Skill panel, select the Intents tab.

Click the Create intent button. Name the intent #Product\_Information, and at a minimum, enter the following example questions to be associated with it.

### Create new dialog node

Now we need to add a node to handle our intent. Click on the Dialog tab, then click on the drop-down menu for the Small Talk node, and select the Add node below option.

Name the node "Ask about product" and assign it our new intent. This means that if Watson Assistant recognizes a user input such as "how do I set the time?", it will direct the conversation to this node.

### Enable webhook from Assistant

Set up access to our Webhook for the IBM Cloud Functions action you created in

Step#4. Select the Options tab:

Enter the public URL endpoint for your action. Return to the Dialog tab, and click on the Ask about product node. From the details panel for the node, click on Customize, and enable Webhooks for this node: Click Apply.

The dialog node should have a Return variable set automatically to \$webhook\_result\_1. This is the variable name you can use to access the result from the Discovery service query.

### Test in Assistant Tooling

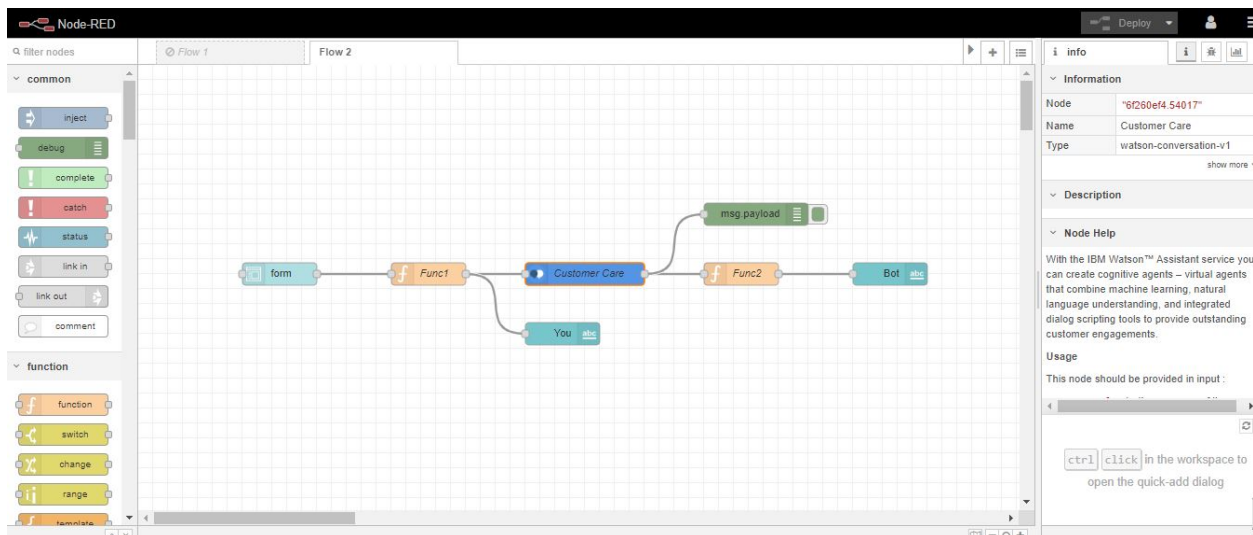
From the Dialog panel, click the Try it button located at the top right side of the panel. Enter some user input: Note that the input "how do I turn on the heater?" has triggered our Ask about product dialog node, which is indicated by the #Product\_Information response. And because we specified that \$webhook\_result\_1.passages be the response, that value is displayed also. You can also verify that the call was successfully completed by clicking on the Manage Context button at

the top right. The response from the Discovery query will be stored in the `$webhook_result_1` variable.

**5. Create flow and configure node:**

At first go to manage palette and install dashboard. Now, Create the flow with the help of following node:

1. Inject
2. Assistant
3. Debug
4. Function
5. Ui\_Form
6. Ui\_Text



**6. Deploy and run Node Red app.**

Deploy the Node Red flow.

Then copy the link url up to `.net/` and paste at anew tab by ui at the end of the url, like this,

<https://node-red-agaft.eu-gb.mybluemix.net/ui>

Customer Care Help-Desk

Chatbot

Enter your input\*

Hello

SUBMIT CANCEL

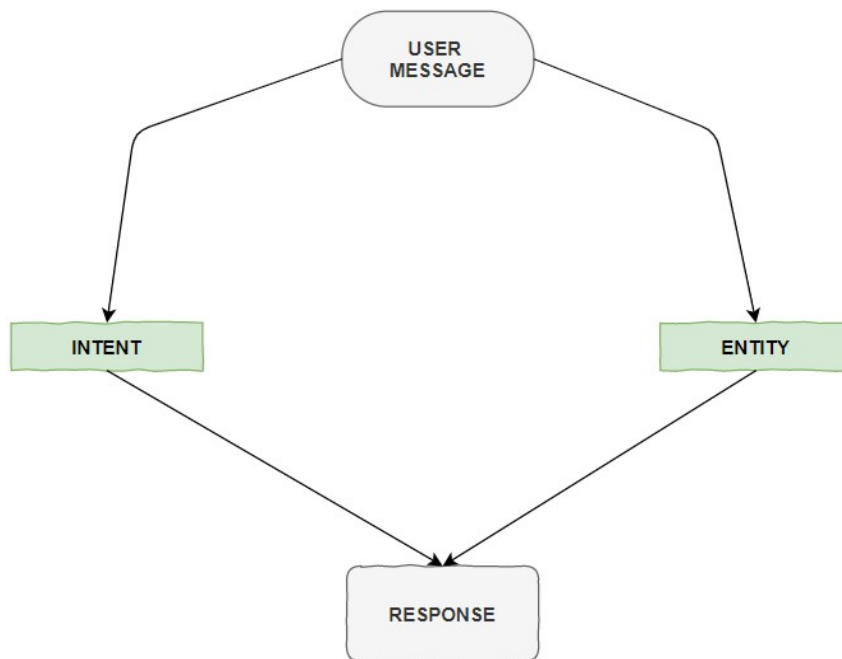
You Hello

Bot

Hello. Good morning

## Theoretical Analysis

### Block Diagram





# Hardware and Software Designing

**Project Requirements:** Python, IBM Cloud, IBM Watson

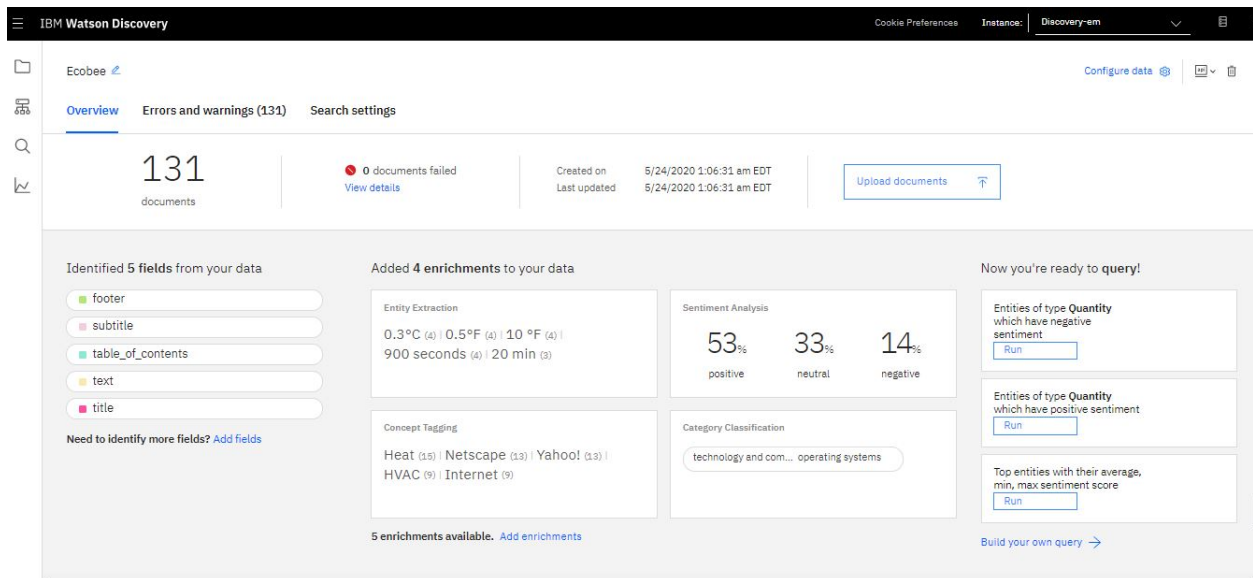
**Functional Requirements:** IBMcloud

**Technical Requirements:** AI, ML, WATSONAI, PYTHON

**Software Requirements:** Watson assistant, Watsondiscovery.

## Experimental Investigations

Watson Discovery:



## Cloud Function:

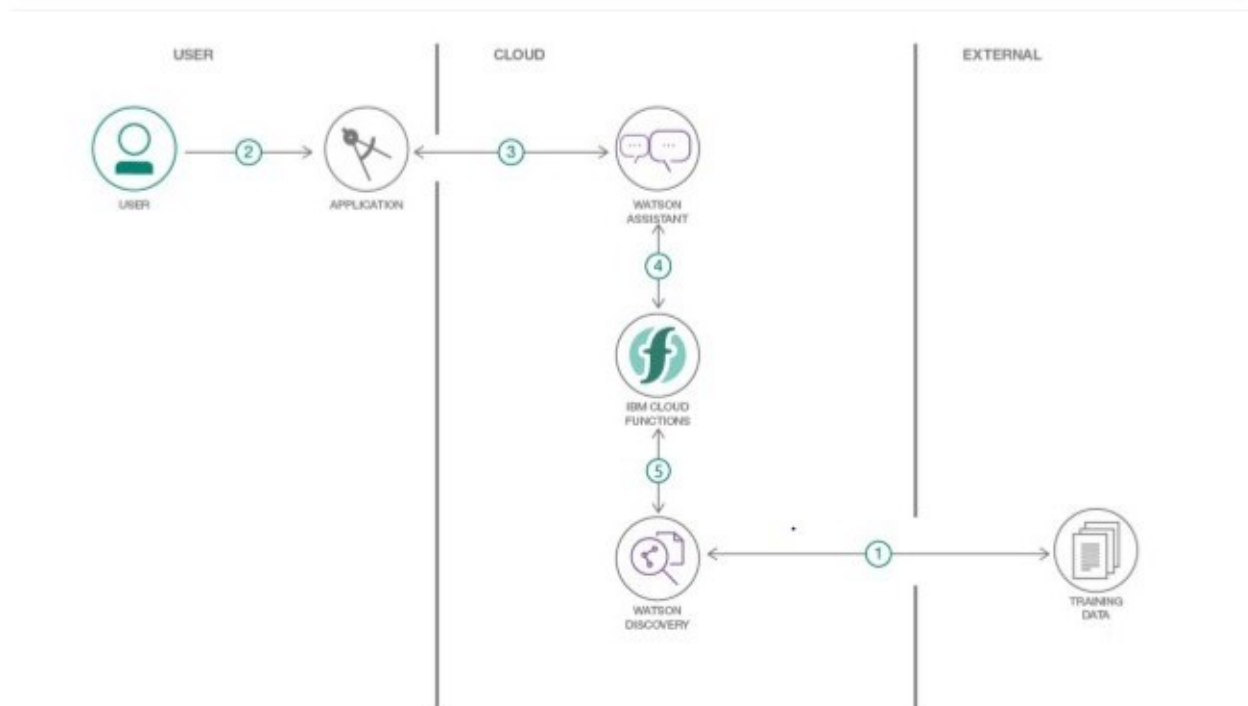
The screenshot shows the IBM Cloud Functions console. The top navigation bar includes the IBM Cloud logo, a search bar, and links to Catalog, Docs, Support, and Manage. The main header shows the function name 'discovery-function' and its namespace '170120116019@git.org.in\_dev(London)'. On the left, a sidebar lists various tabs: Code, Parameters, Runtime, Endpoints, Connected Triggers, Enclosing Sequences, and Logs. The 'Code' tab is selected, displaying a Node.js 10 function. The code is a web action that uses the 'discovery-v1' API to query a collection based on environment ID, collection ID, and a natural language query. The code is as follows:

```
1 // ...  
16  
17 const assert = require('assert');  
18 const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');  
19  
20 // ...  
29 * function main(params) {  
30 *   return new Promise(function (resolve, reject) {  
31  
32     let discovery;  
33  
34     if (params.iam_apikey) {  
35 *       discovery = new DiscoveryV1({  
36 *         'iam_apikey': params.iam_apikey,  
37 *         'url': params.url,  
38 *         'version': '2019-03-25'  
39 *       });  
40 *     }  
41 *     else {  
42 *       discovery = new DiscoveryV1({  
43 *         'username': params.username,  
44 *         'password': params.password,  
45 *         'url': params.url,  
46 *         'version': '2019-03-25'  
47 *       });  
48 *     }  
49  
50 *     discovery.query({  
51 *       'environment_id': params.environment_id,  
52 *       'collection_id': params.collection_id,  
53 *       'natural_language_query': params.input,  
54 *     });  
55 *   });  
56 * }
```

## Watson Assistant:

The screenshot shows the IBM Watson Assistant console. The top navigation bar includes the IBM Watson Assistant Lite logo and an 'Upgrade' button. The main header shows the skill name 'Customer Care Sample Skill' and a 'Save new version' button. On the left, a sidebar lists various tabs: Intents, Entities, Dialog, Options, Analytics, Versions, and Content Catalog. The 'Dialog' tab is selected, displaying a dialog flow. The flow starts with an 'Opening' node, followed by a 'What are your hours?' node, then a 'Where are you located?' node, then an 'I want to make an appointment' node, then a '#General\_Greetings' node, and finally a '#Goodbye' node. The right sidebar shows a 'Try it out' section with a chat interface. The chat interface displays a welcome message and a prompt to set a temperature for a heater. The prompt is: 'How to set a temperature of the heater?'. The chat interface also shows a list of intents: '#Product\_Information', '#General\_Greetings', and '#Goodbye'. The chat interface includes a search bar and a 'Use the up key for most recent' button.

# FLOWCHART



1. The document is annotated using Watson DiscoverySDU
2. The user interacts with the backend server via the app UI. The front end app UI is a chat bot that engages the user in a conversation.
3. Dialog between the user and backend server is coordinated using a Watson Assistant dialog skill.
4. If the user asks a product operation question, a search query is passed to a predefined IBM Cloud Functionsaction.
5. The Cloud Functions action will query the Watson Discovery service and return the results.

## Results

The chat bot was successfully made using Watson assistant and using SDU. All the services were integrated using Node Red Application.

Customer Care Help-Desk

Chatbot

Enter your input  
How to set temprature of the heater

SUBMIT

CANCEL

You

How to set temprature of the heater

Bot

"Depending on your system, one or more the following options are shown: ☐  
Cool: Turn on the air conditioner when the current temperature rises above the  
set temperature. ☐ Heat: Turn on the heat when the current temperature drops  
below the set temperature. ☐ Auto: Activate the heating or cooling system as  
required to keep your home"

# Advantages and Disadvantages

## Advantages:

- A. Reduced costs: Chatbots eliminate the need for labor during online interaction with customers. This is obviously a great advantage for companies that receive multiple queries at once. In addition to saving costs with them, companies can align the chat bot with their objectives, and use them as a means to enhance customer conversion.
- B. 24/7 Availability: Unlike humans, once we install a chat bot, it can handle queries at any time of day. Thus, the customer does not have to wait for a commercial of the company to help him. This also allows companies to monitor customer « traffic » during non-working hours and contact them later.
- C. Learning and updating: AI-based chatbots are able to learn from interactions and update independently. This is one of the main advantages. When you hire a new employee, you have to train them continuously. However, chatbots « form » themselves (with certain limitations, of course).
- D. Management of multiple clients: Humans can serve a limited number of customers at the same time. This restriction does not exist for chatbots, and they can manage all the necessary queries simultaneously. This is one of the main advantages of using chat bot, as no customer is left unattended and you are solving different problems at the same time. There are chatbots companies already working on developing voice chat bot services.

## Disadvantages:

- A. Complex interface: It is often considered that chatbots are complicated and need a lot of time to understand what you want in customer. Sometimes, it can also annoy the client about their slowness, or their difficulty in filtering responses. They don't get you right: Fixed chatbots can get stuck easily. If a query doesn't relate to something you've previously « taught » it, you won't understand it. This can lead to a frustrated customer and the loss of the sale. Other times they do understand you, but they need double (or triple) as many messages as one person, which spoils the user experience.
- B. Bad memory: The chatbots are not able to memorize a conversation already had, which

forces the user to write the same thing over and over again. This can be cumbersome for the client and annoying for the effort required. Therefore, it is important to be careful when designing chatbots and make sure that the program is able to understand users' queries and respond accordingly.

# Applications

Some applications can be: -

1. **Help User:** This chat bot will be useful for the user to ask the assistant the queries related to the Product and will give them clear guidance about the Product. If the Assistant doesn't know about a certain query, it will redirect to the correct person for it.
2. **Content delivery:** Media Publishers have realized that chatbots are a powerful way to engage with their audiences and monitor engagement to gain valuable insights on reader interests. Chat with the CNN and Wall Street Journal Chatbots on Facebook Messenger and receive the latest news directly in Messenger, without having to visit their websites.
3. **Companionship:** The primary function of the chat bot is to be a virtual companion – To speak with senior people on general topics like the weather, nature, hobbies, movies, music, news, etc. The chat bot asks questions, reacts to the answers, is able to speak on various topics, and share interesting news and facts from Google.

## **Conclusion**

This chat bot will be useful for the user to ask the assistant the queries related to the Product and will give them clear guidance about the product. If the Assistant doesn't know about a certain query, it will redirect to the correct person for it. Chatbots are quickly making transformational changes and allowing businesses to thrive through customer interactions. The feedback and survey through chatbots strengthen the position of businesses as they analyze the reason behind different levels of customer approval. Use of conversational AI chatbots only means better engagement and relentless need for customer satisfaction in the near future.

## **Future Scope**

Future Scope of this chat bot can be by adding the following to make it more advance: -

1] Smarter Virtual Assistants: Much of what virtual assistants do now are basic skills, such as retrieving data and basic computation. As natural language processing (NLP) continues to mature, virtual assistants will improve their comprehension and response capabilities, allowing for their use to become more widespread and complex. Also, as machine learning progresses, we may see virtual assistants become smarter and begin to learn and predict customer needs.

2] Integration with IoT Devices: Car speakers, smart home devices, and wearables are just a few examples where the virtual assistant is departing from its original hardware and making its way to in-context devices. These integrations ensure that virtual assistants can always be near their human and ready to support any need. It is expected that these integrations will continue at an accelerated pace throughout 2018.

3] Voice-control: Voice recognition can be added with the virtual assistant. Then the customer can control application by using his voice. Soon, we could be joining meetings with a voice command, instead of dialing in the long meeting ID and password.



## Bibliography

1. [https://www.ibm.com/cloud/architecture/tutorials/cognitive\\_discovery](https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery)
2. <https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started>
3. <https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/>
4. <http://www.iotgyan.com/learning-resource/integration-of-watson-assistant-to-node-red>
5. <https://github.com/IBM/watson-discovery-sdu-with-assistant>
6. [https://www.youtube.com/watch?v=\\_qJ-SxuxXUM](https://www.youtube.com/watch?v=_qJ-SxuxXUM)