Project Report

On

Intelligent Customer Helpdesk with Smart Document Understanding

in

Artificial Intelligence

by

Aanchal Shrivastava

(shrivastavaaanchal34@gmail.com)

1. Introduction

1.1 Overview

We will assemble a Chabot that utilizes different Watson AI Services (Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-Red) to convey a compelling Web based UI through which we can chat with the assistant.

We will integrate the Watson Discovery service with Watson Assistant using webhooks.

- Project Requirements :Node-RED,IBMCloud,IBMWatson,NodeJS
- Functional Requirements : IBMCloud
- TechnicalRequirements:AI,ML,WatsonAI,NodeJS
- Software Requirements : Watson Assistant, Watson Discovery, Watson Cloud Functions,Node-RED
- Project Deliverables : Intelligent Chatbot with Smart Document Understanding
- Project Team : Aanchal Shrivastava
- Project Duration : 19Days

1.2 Purpose

The typical customer care Chabot 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 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. So unless and until customer specifically asks for a customer representative the bot will try to solve all your queries.

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 owners manual is important and what is not. This will improve the answers returned from the queries. Then using Watson actions as Webhook, WatsonDiscovery can be integratedwithWatsonassistant.FinallyusingNode-Red,WatsonAssistant can be integrated with a WebUI. This UI cant henbe used to connect with Watson assistant and chat with it.

1.2.1 Scope of Work

- Create a customer care dialog skill inWatsonAssistant.
- 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
- Build a web application with integration to all these services & deploy the same on IBM Cloud Platform

2. Literature Survey

2.1 Existing Problem

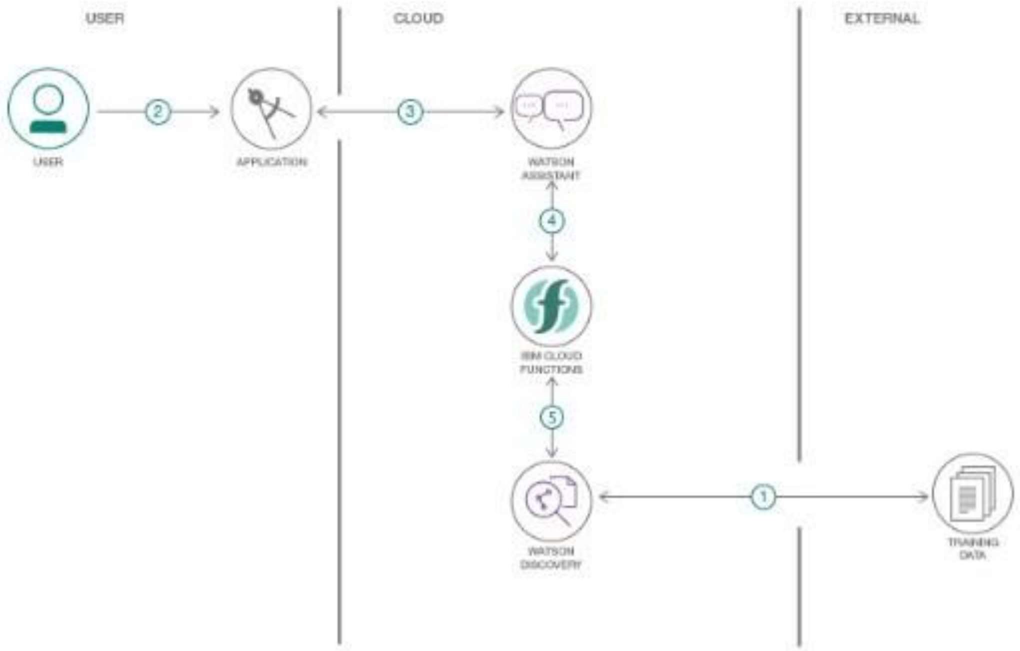
The typical customer care Chabot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn't valid or offer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

2.2 Proposed Solution

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 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. So unless and until customer specifically asks for a customer representative the bot will try to solve all your queries.
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 owners manual is important and what is not. This will improve the answers returned from the queries. Then using Watson actions as webhook,Watson DiscoverycanbeintegratedwithWatsonassistant.FinallyusingNode-Red, Watson assistant can be integrated with a webUI.This UI can then be used to connect with Watson assistant and chat with it.

3. Theoretical Analysis

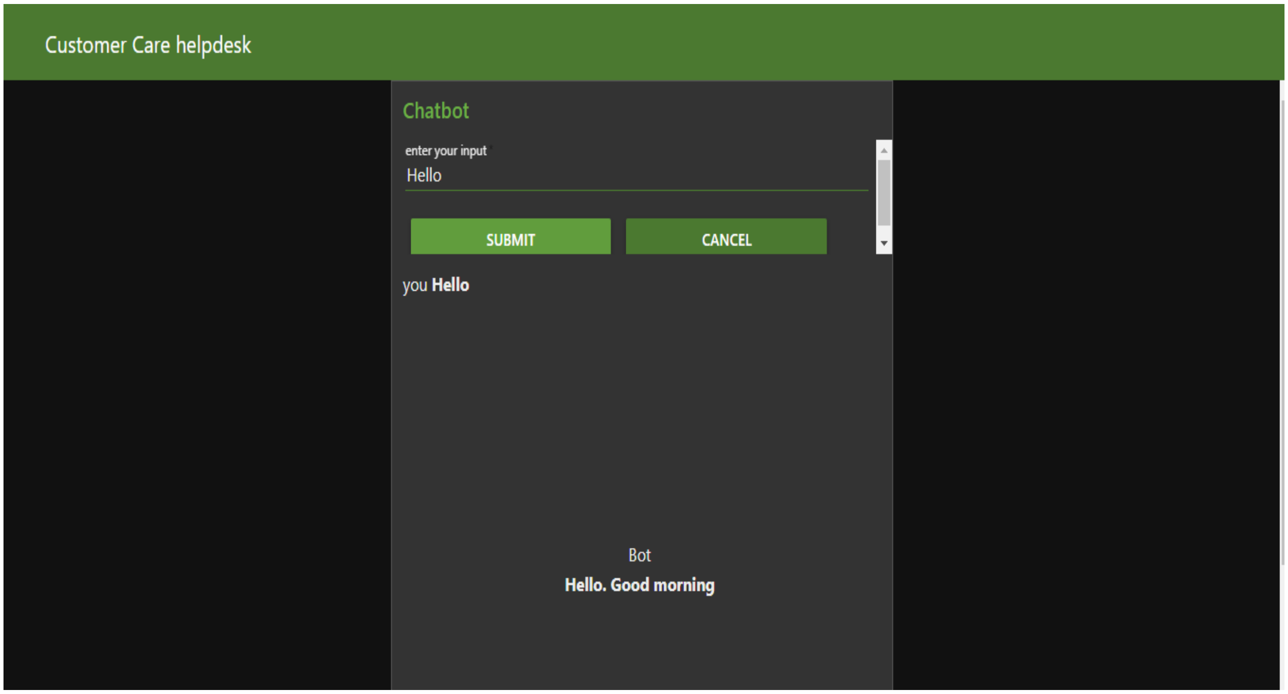
3.1 Block / Flow diagram



3.2 Hardware /Software Designing

- Create necessary WatsonServices.
- Configure WatsonDiscovery.
- Create Watson Cloud FunctionsAction.
- Configure WatsonAssistant.
- Integrate Watson Discovery with Watson Assistant usingwebhook.
- Build Node-REDflow to integrate Watson Assistant and Web Dashboard.

4. Experimental Investigation

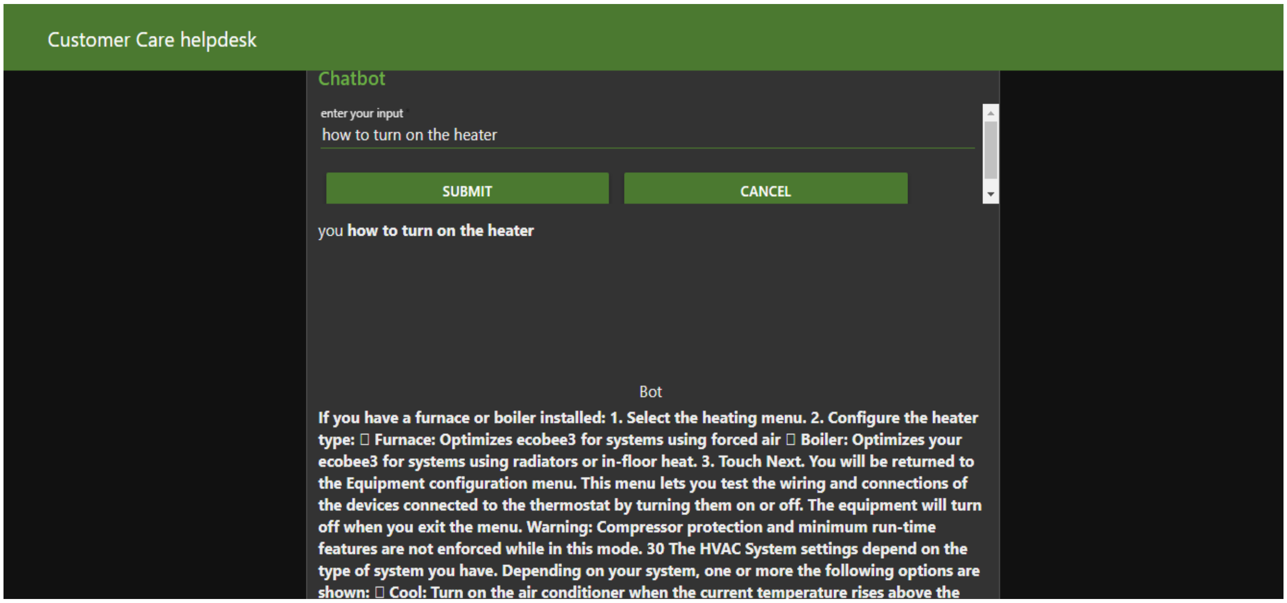


enter your input: Hello

you **Hello**

Bot

Hello. Good morning

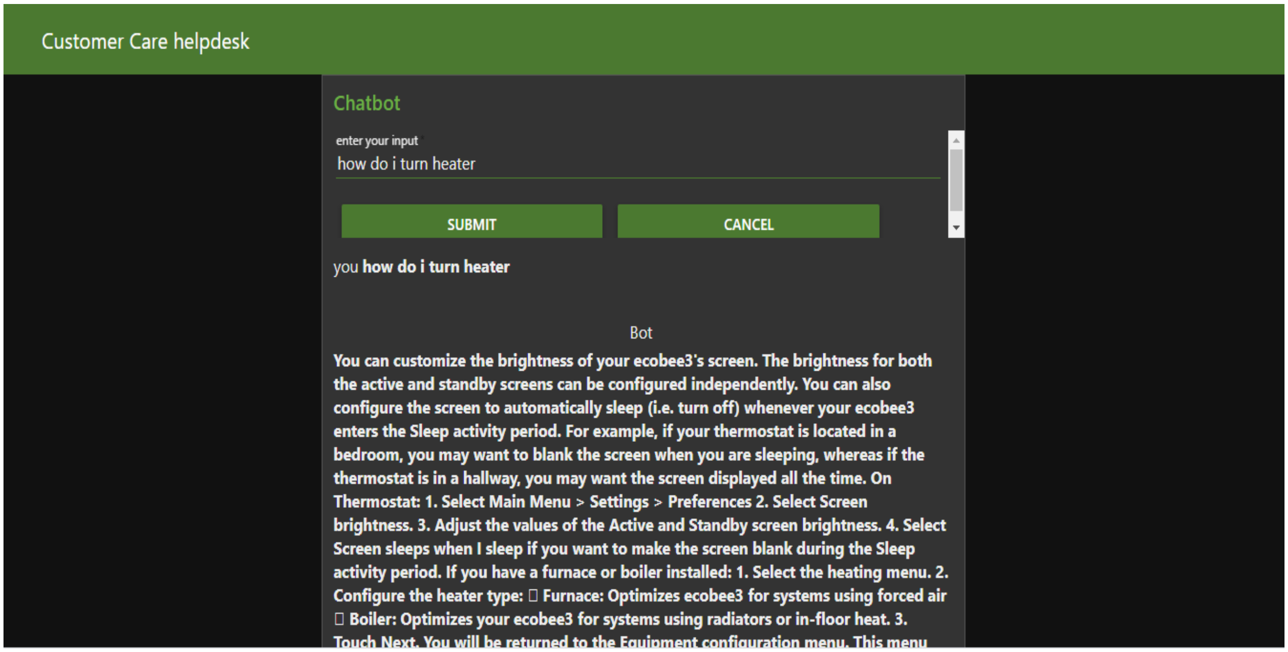


enter your input: how to turn on the heater

you **how to turn on the heater**

Bot

If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: □ Furnace: Optimizes ecobee3 for systems using forced air □ Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3. Touch Next. You will be returned to the Equipment configuration menu. This menu lets you test the wiring and connections of the devices connected to the thermostat by turning them on or off. The equipment will turn off when you exit the menu. Warning: Compressor protection and minimum run-time features are not enforced while in this mode. 30 The HVAC System settings depend on the type of system you have. 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



enter your input *
how do i turn heater

you **how do i turn heater**

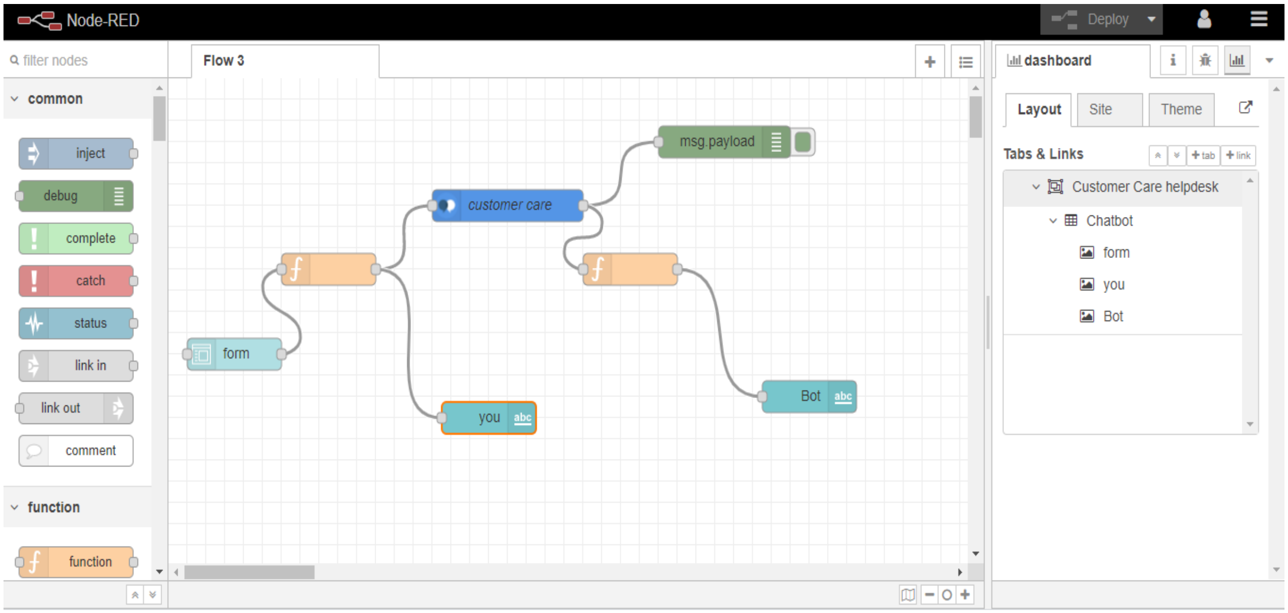
Bot

You can customize the brightness of your ecobee3's screen. The brightness for both the active and standby screens can be configured independently. You can also configure the screen to automatically sleep (i.e. turn off) whenever your ecobee3 enters the Sleep activity period. For example, if your thermostat is located in a bedroom, you may want to blank the screen when you are sleeping, whereas if the thermostat is in a hallway, you may want the screen displayed all the time. On Thermostat: 1. Select Main Menu > Settings > Preferences 2. Select Screen brightness. 3. Adjust the values of the Active and Standby screen brightness. 4. Select Screen sleeps when I sleep if you want to make the screen blank during the Sleep activity period. If you have a furnace or boiler installed: 1. Select the heating menu. 2. Configure the heater type: □ Furnace: Optimizes ecobee3 for systems using forced air □ Boiler: Optimizes your ecobee3 for systems using radiators or in-floor heat. 3. Touch Next. You will be returned to the Equipment configuration menu. This menu

5. Flowchart

Insert the following nodes into the flow in Node-RED.

- Debug
- ui_Form
- ui_Text
- Function
- Assistant



6. Results

Web based UI was developed by integrating all the services using Node-RED.

URL for UI Dashboard :- https://node-red-xisaa.eu-gb.mybluemix.net/ui/

7. Advantages &Disadvantages

Advantages -

- Reduces Man Power
- Cost Efficient
- Less and less calls will be diverted to Customer Representatives.

Disadvantages -

- Sometimesitcanmisleadcutomersasittriestosearchirrelevantinformation in the manual.
- It may also give same answers to different queries.

8. Applications

This Chabot can be deployed to various websites as it can solve a lot of basic questions.

It can be used to deploy as Customer Helpdesk for small scale products as their manual usually has the solution for the user's problems.

9. Conclusion

An Intelligent Customer Helpdesk Chabot was created using various Watson services like Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-RED.

10. Future Scope

In the future, various other Watson services likeText-To-SpeechandSpeech-To-Textcanbeintegratedinthechatbot.This can make the Chabot Hands-free.

11. Bibliography

Appendix

Source Code

Node-Red Flow Code -

```
[
{
"id": "52825875.ee8c68",
"type": "tab",
"label": "Flow 3",
"disabled": false,
"info": ""
},
{
"id": "62aa25c6.6a333c",
"type": "ui_form",
"z": "52825875.ee8c68",
"name": "",
"label": "",
"group": "21762bef.6437e4",
"order": 1,
 "width": 14,

 "height": 2,
"options": [

 {

 "label": "enter your input",

 "value": "input",

 "type": "text",

 "required": true,

 "rows": null

 }

 ],

 "formValue": {

 "input": ""

 },

 "payload": "",

 "submit": "submit",

 "cancel": "cancel",

 "topic": "",

 "x": 70,

 "y": 260,

 "wires": [
[
"13558c61.d69904"
]
]
},
{ "id": "13558c61.d69904",
"type": "function",
"z": "52825875.ee8c68",
"name": "",
"func": "msg.payload = msg.payload.input;\nreturn msg;",
"outputs": 1,
"noerr": 0,
"x": 170,
"y": 180,
"wires": [
[
"cd3f0b21.68e9c8",
"b94e8d69.46abe"
]
]
```

```
},
{
"id": "da3959ee.949338",
"type": "function",
"z": "52825875.ee8c68",
"name": "",
"func": "msg.payload.text=\"\";\nif(msg.payload.context.webhook_result_1){\n for(var i in msg.payload.context.webhook_result_1.results){\n msg.payload.text=msg.payload.text+\"\\n\"+msg.payload.context.webhook_result_1.results[i].text;\n}\n msg.payload=msg.payload.text;\n}\n\nelse\nmsg.payload = msg.payload.output.text[0];\nreturn msg;\n\n",
"outputs": 1,
"noerr": 0,
"x": 490,
"y": 180,
"wires": [
[
"ee1099ac.baabc8"
]
]
```

```
 },
 {
 "id": "cd3f0b21.68e9c8",
 "type": "watson-conversation-v1",
 "z": "52825875.ee8c68",
 "name": "customer care",
 "workspaceid": "c13d047e-f08f-42a4-8b03-92b29d7a7cee",
"multiuser": false,
 "context": false,
 "empty-payload": false,
 "service-endpoint": "",
 "timeout": "",
 "optout-learning": false,
 "x": 360,
 "y": 120,
 "wires": [
 [
 "5b9567d0.fd50e8",
 "da3959ee.949338"
 ]
 ]
 },
 {
 "id": "b94e8d69.46abe",
 "type": "ui_text",
 "z": "52825875.ee8c68",
 "group": "21762bef.6437e4",
 "order": 2,
 "width": 14,
 "height": 1,
 "name": "",
"label": "you",
 "format": "{{msg.payload}}",
 "layout": "row-left",
 "x": 340,
 "y": 320,
 "wires": []
 },
 {
 "id": "5b9567d0.fd50e8",
 "type": "debug",
 "z": "52825875.ee8c68",
"name": "",
"active": true,
"tosidebar": true,
"console": false,
"tostatus": false,
"complete": "false",
"x": 590,
"y": 60,
"wires": []
},
{
"id": "ee1099ac.baabc8",
"type": "ui_text",
"z": "52825875.ee8c68", "group": "21762bef.6437e4",
"order": 3,
"width": 14,
"height": 9,
"name": "",
"label": "Bot",
"format": "{{msg.payload}}",
"layout": "col-center",
"x": 680,
"y": 300,
"wires": []
},
{
"id": "21762bef.6437e4",
"type": "ui_group",
"z": "",
"name": "Chatbot",
"tab": "99923adf.1ef448",
"order": 1,
"disp": true,
"width": 14,
"collapse": false
},
{ id": "99923adf.1ef448",
"type": "ui_tab",
"z": "",
"name": "Customer Care helpdesk",
 "icon": "dashboard",
 "disabled": false,
 "hidden": false
 }
 ]
```

Watson Cloud Function Action Code

```
/**
*
* @param {object} params
* @param {string} params.iam_apikey
* @param {string} params.url
* @param {string} params.username
* @param {string} params.password
* @param {string} params.environment_id
* @param {string} params.collection_id
* @param {string} params.configuration_id
* @param {string} params.input
*
* @return {object}
*
*/


const assert = require('assert');
const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');


/**
* * main() will be run when you invoke this action
*
* @param Cloud Functions actions accept a single parameter, which must be a JSON object.
*
* @return The output of this action, which must be a JSON object.
*
*/
function main(params) {
return new Promise(function (resolve, reject) {

let discovery;

if (params.iam_apikey){
discovery = new DiscoveryV1({
'iam_apikey': params.iam_apikey,
'url': params.url,
'version': '2019-03-25'
});
}
else {
discovery = new DiscoveryV1({
'username': params.username,
'password': params.password,
'url': params.url,
'version': '2019-03-25'
});
}

discovery.query({
'environment_id': params.environment_id,
'collection_id': params.collection_id,
'natural_language_query': params.input,
'passages': true,
'count': 3,
'passages_count': 3
}, function(err, data) {
if (err) {
return reject(err);
}
return resolve(data);
});
});
}
```

References :-

1. 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=Jpr3wVH3FVA