PROJECT REPORT

Intelligent Customer Help Desk with Smart

Category: Artificial Intelligence Developer

Internship at SmartInternz

Name: Dipika Shivaji Aade.

Id: dipikaaade0@gmail.com

CONTENTS

1.INTRODUCTION

1.1 Overview

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 owner's manual that has been uploaded into Watson Discovery.

1.2 Purpose

The purpose is to Enhance the customer helpdesks with Smart Document Understanding using webhooks in Watson Assistant.

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.

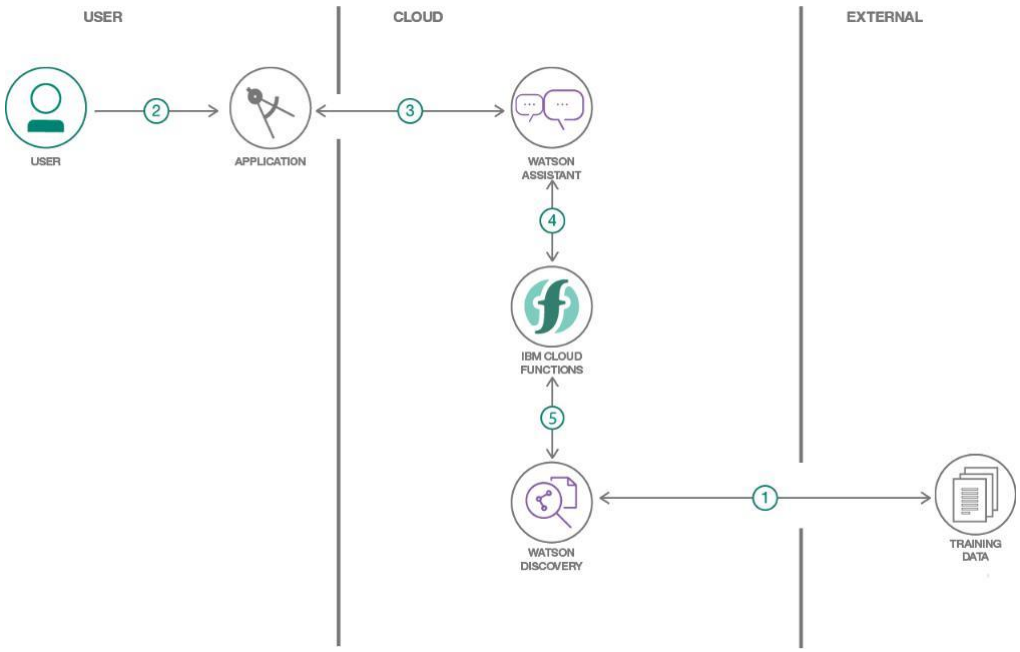
2.2 Proposed solution

In this project, 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.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

1. Create IBM Cloud Services
2. Configure Watson Discovery
3. Create IBM Cloud Functions action
4. Configure Watson Assistant
5. Build Node-RED Flow to Integrate All Services
6. Configure the nodes and Build A Web Dashboard in Node-RED
7. Deploy and Run the application

4. Experimental Investigation

1) Create IBM Cloud Services

a. Watson Discovery

b. Watson Assistant

c. Node Red

2) Configure Watson Discovery

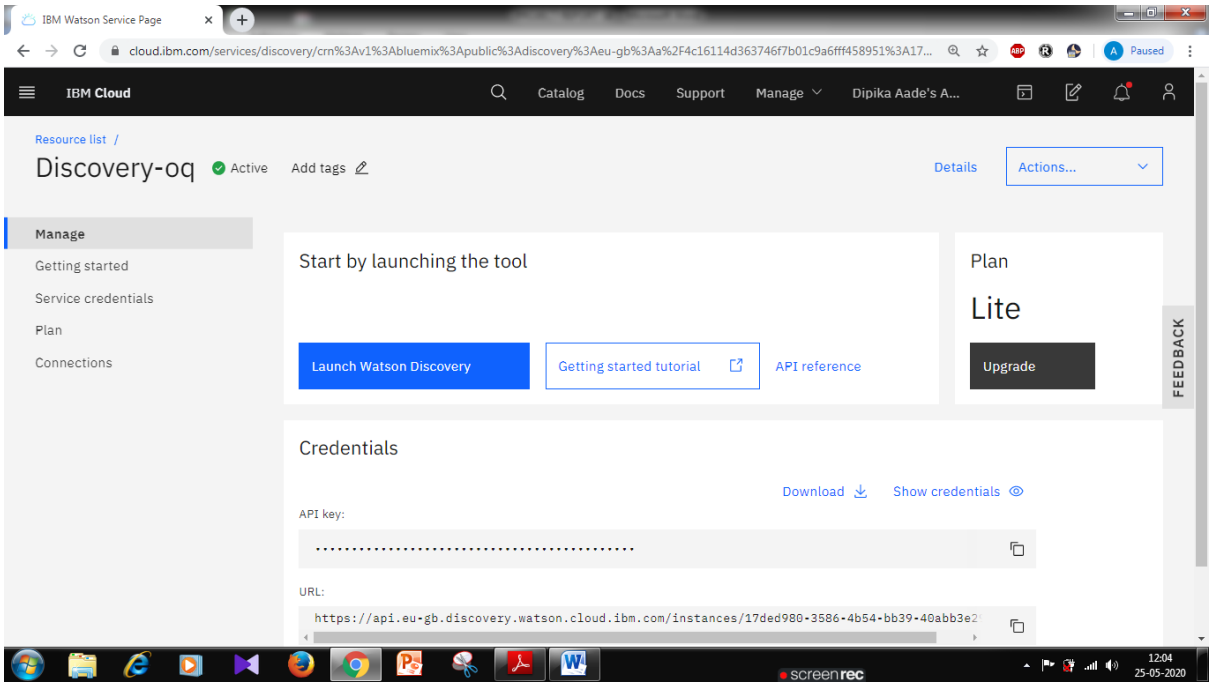
After creating and launching the discovery from the Catalog, Import the document on which on which we need to train the discovery service. We have selected the ecobee3 user guide located in the data directory of our local repository.
The Ecobee3 is a popular residential thermostat that has a WIFI interface and multiple configuration options. The result of the queries performed without configuring the data present in the document won't be that accurate. But the results improve significantly after applying SDU (Smart Document Understanding).

This can be done easily by clicking on the configure setting and then labelling each word or element present in the document as their respective label such as title, subtitle, text, image and Footer. Some of the labels are not present in the lite plan.

In the lite plan we are provided with limited content of IBM Watson, the labels help us in segmentation of the document which helps the discovery to understand the document better and provide better results. The results provided by the discovery can be improved, all the results are shown in assistant in which the discovery finds the sentiment to be positive i.e. matching between the question or query entered by the user and the data of the document. Better the sentiment analysis accurate the results are.

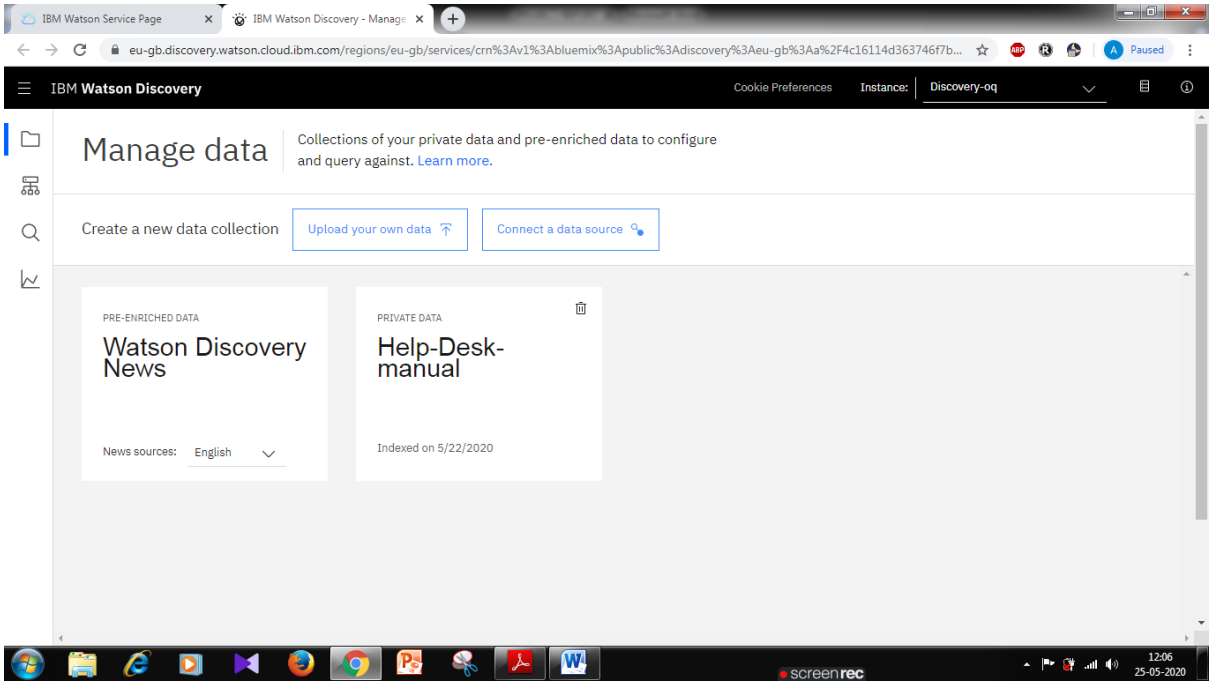
Follow the below mentioned steps:

• After creating the discovery from the catalog, we will be redirected to this base page of discovery where the name of the discovery along with its API Key and URL are mentioned. These credentials will be used in further steps.

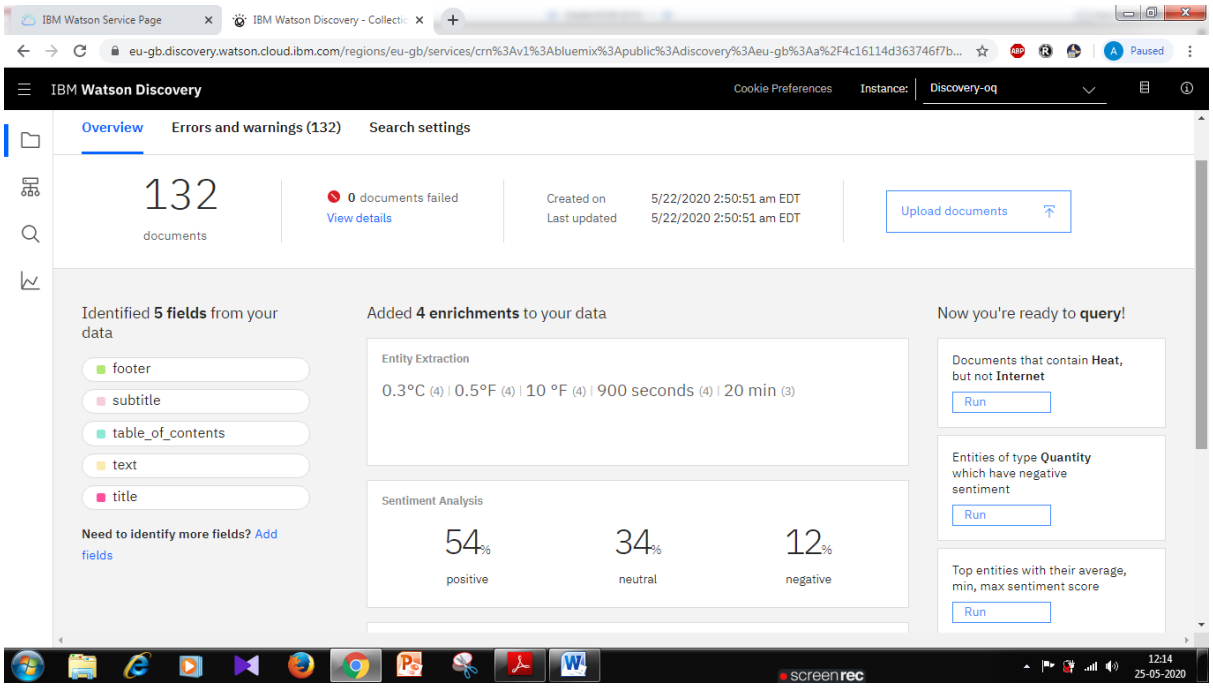


Click on the Launch Watson Discovery [1] to launch the discovery.

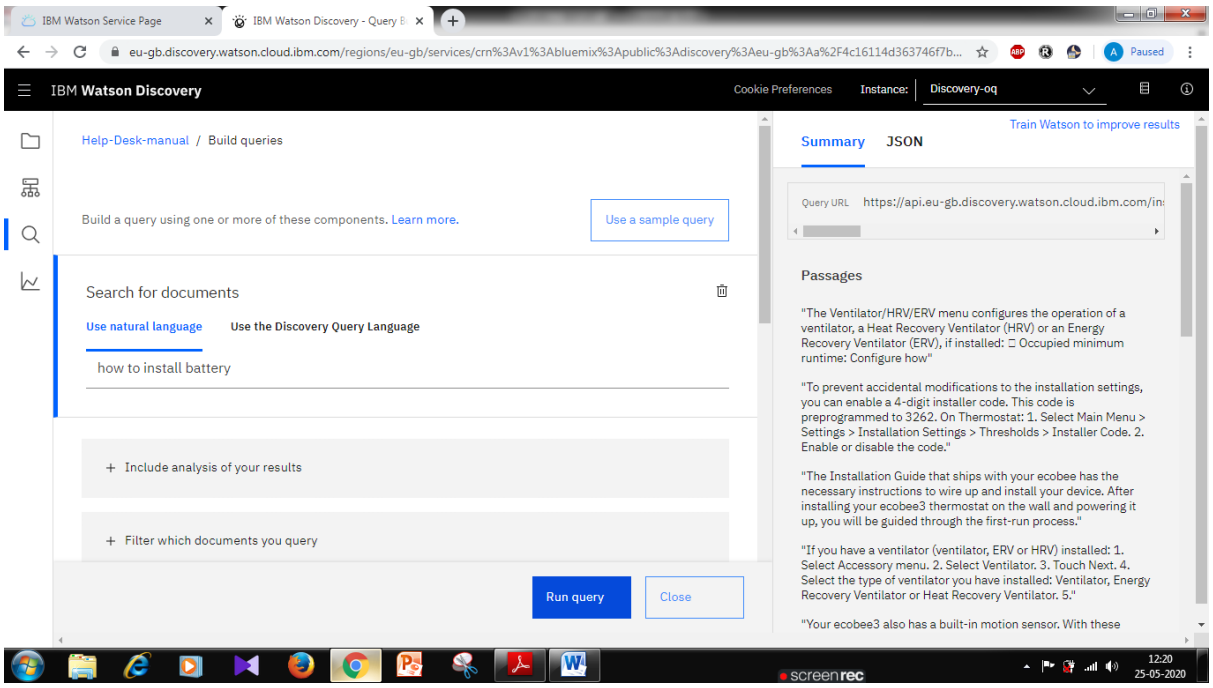
• Now in the next step we have to upload the data by clicking, upload your data. Here we have already uploaded the data as manual.


IBM Watson Service Page
IBM Watson Discovery - Manage
eu-gb.discovery.watson.cloud.ibm.com/regions/eu-gb/services/crn%3Av1%3Abluemix%3Apublic%3Adiscovery%3Aeu-gb%3Aa%2F4c16114d363746f7b...
Paused
IBM Watson Discovery
Cookie Preferences
Instance:
Discovery-oq
Manage data
Collections of your private data and pre-enriched data to configure and query against. Learn more.
Create a new data collection
Upload your own data
Connect a data source
PRE-ENRICHED DATA
Watson Discovery News
News sources:
English
PRIVATE DATA
Help-Desk-manual
Indexed on 5/22/2020
screenrec
12:06
25-05-2020

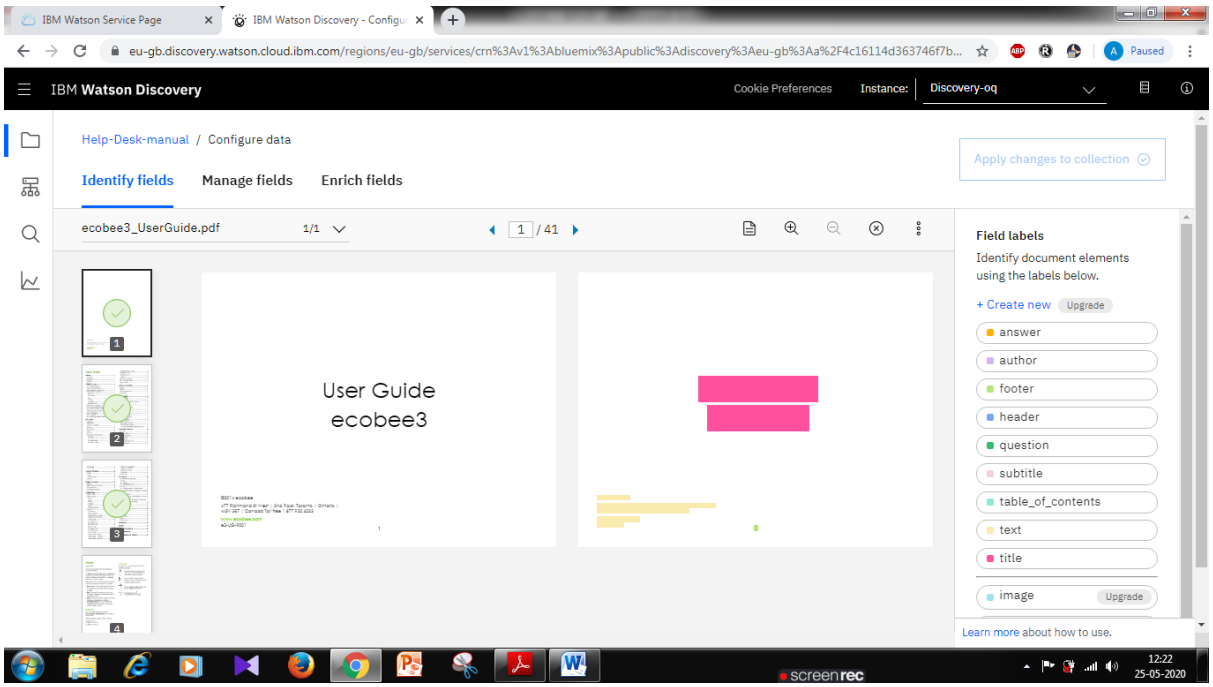
After uploading the document we will see the page like this, we can ignore the warnings section as it is due to the normalization process and is of no worry. Now click on the build your own query [1] , so that we can have an idea how the results have significantly improved after configuring the data.



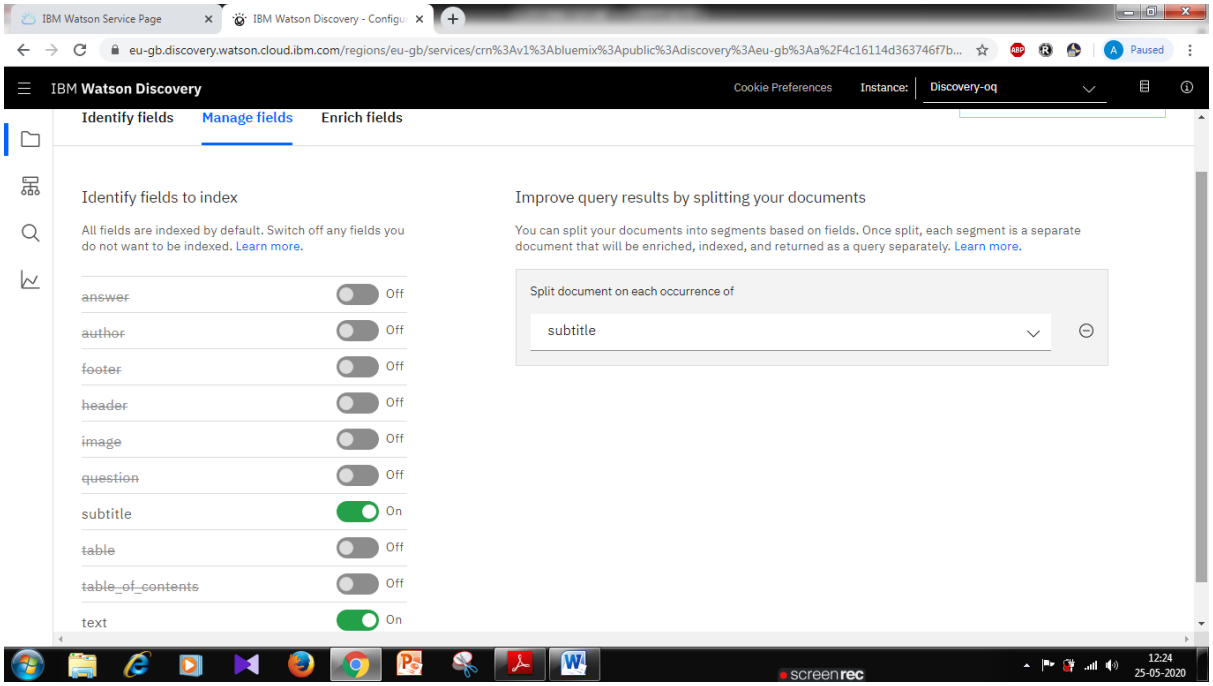
Enter the queries related to thermostat and view the results. As you will see, the results are not very useful and not even that relevant. The next step is to annotate the document with SDU.



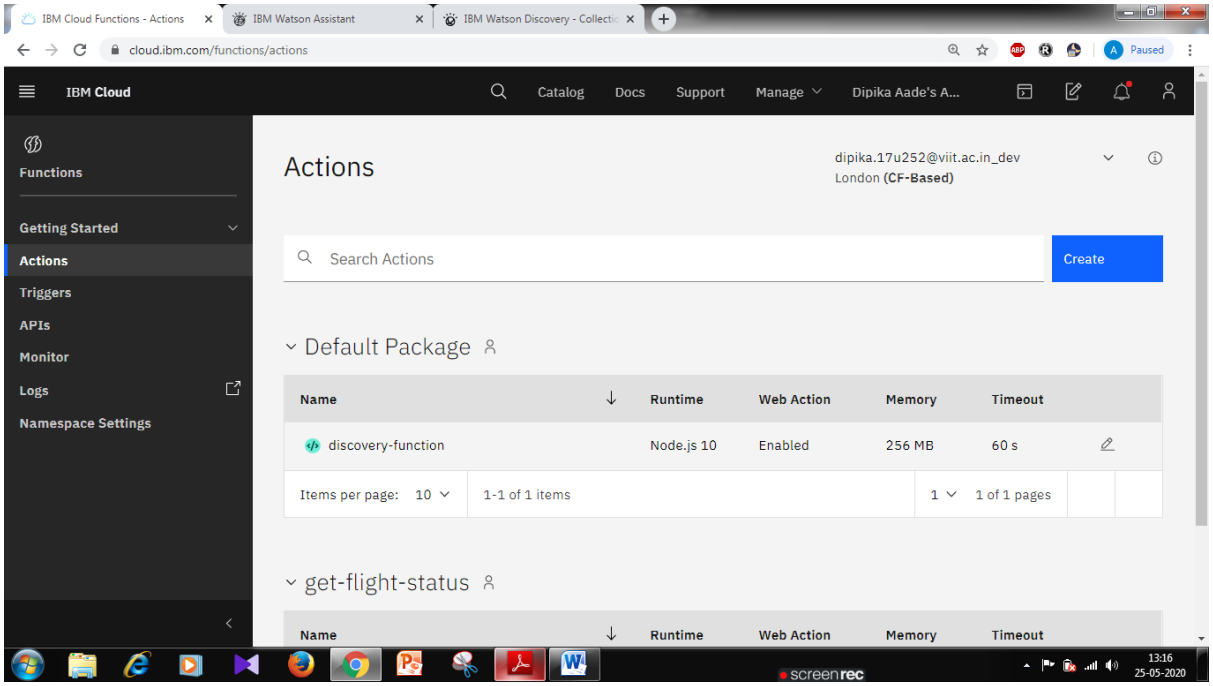
Below is the layout of Identify fields tab of the SDU annotation panel:



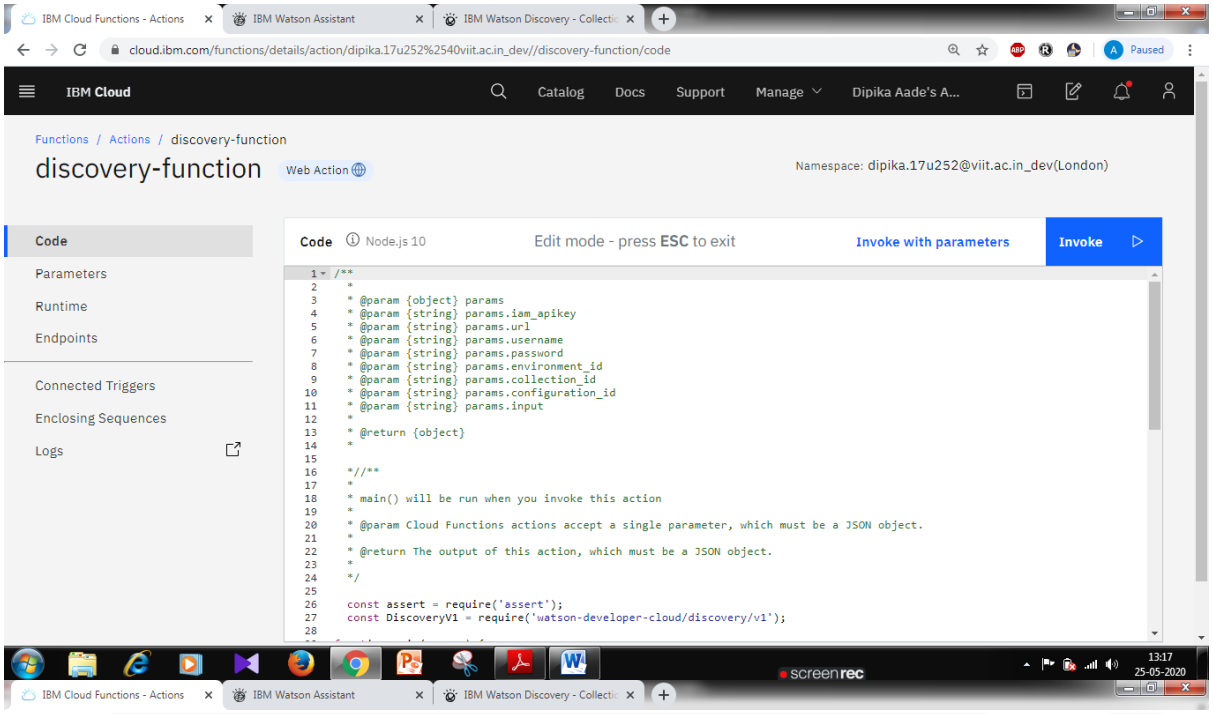
For further segmentation and making the sub documents, we have to manage the fields. Here we are provided with the option of identifying field to index i.e. what all texts are important for us, as we can see in the below snip, we have turned on only subtitle and text because they are the only 2 labels in which we are interested. On the right side we have the option splitting the document as per choice of our label. We have selected subtitle here. This can vary as per different needs of user.

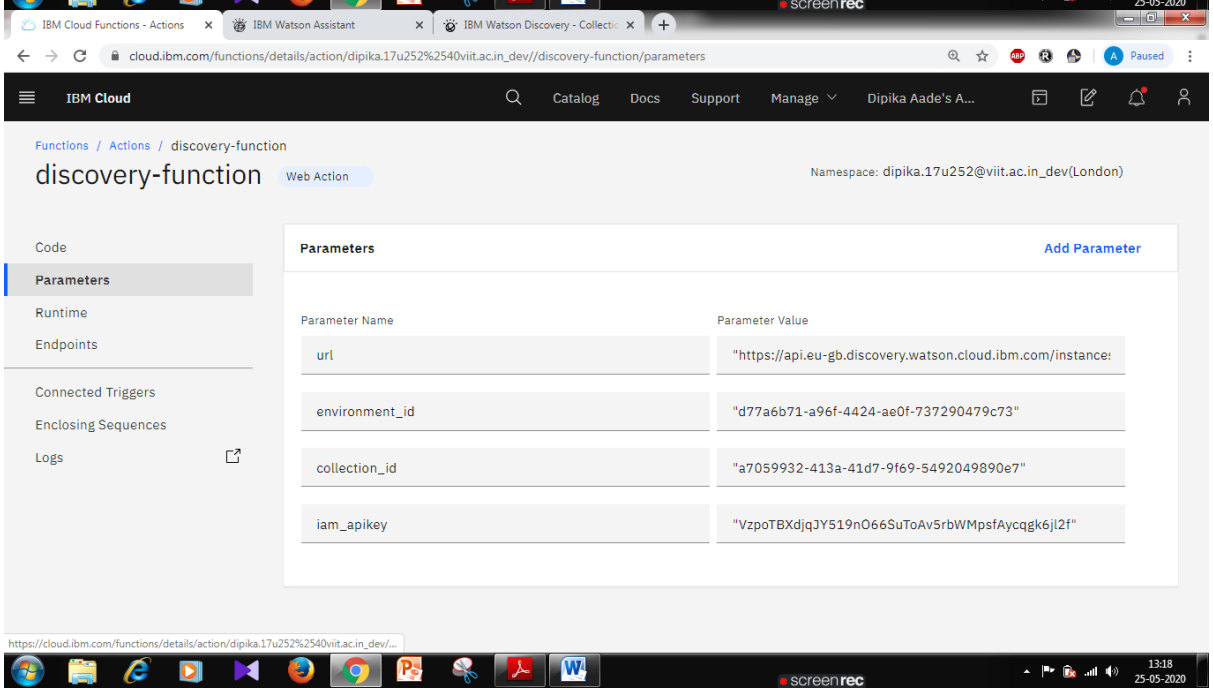


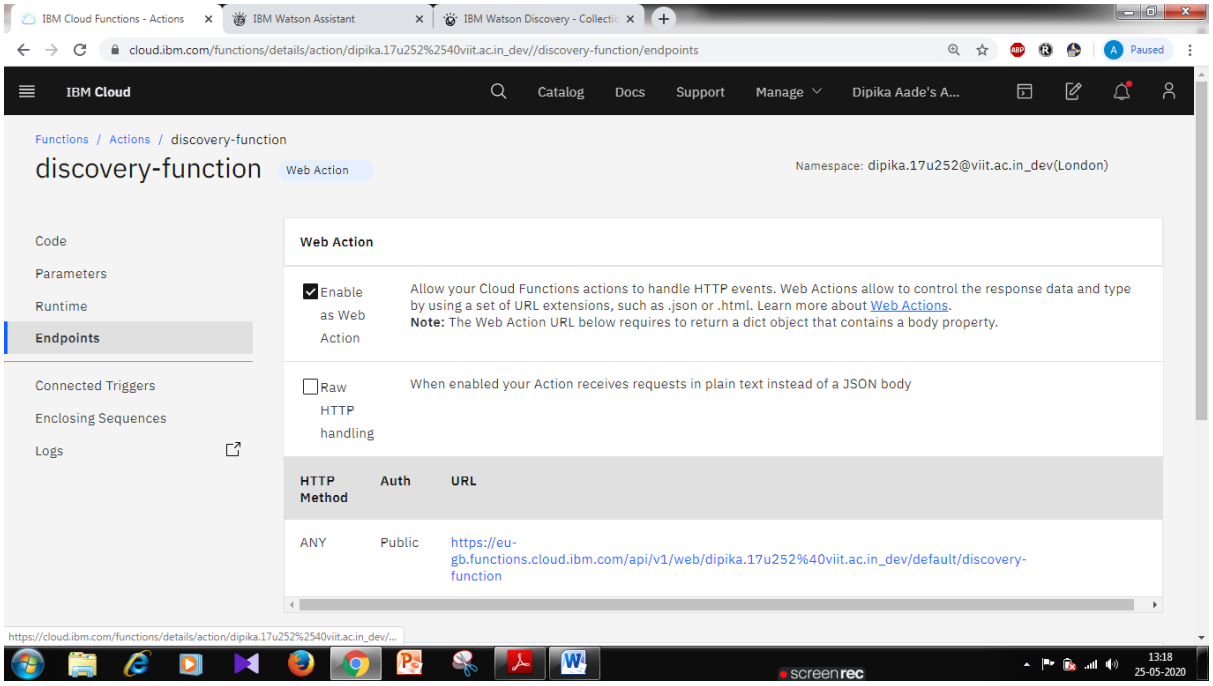
It is used to link the discovery with assistant, so that our queries can be answered by the discovery. After selecting the action from the IBM catalog, we have to click on the action tab as shown on the left menu. Here we made the Information function. Then we can post the code which will help us to link the discovery.



We can make the parameters as per the code and paste the parameter value from the discovery credentials. After that we have to click on the endpoint and enable the web action which will generate a public URL and it will be further used.


IBM Cloud Functions - Actions
IBM Watson Assistant
IBM Watson Discovery - Collectio
cloud.ibm.com/functions/details/action/dipika.17u252%2540viit.ac.in_dev//discovery-function/code
Paused
IBM Cloud
Catalog
Docs
Support
Manage
Dipika Aade's A...
Functions / Actions / discovery-function
discovery-function
Web Action
Namespace: dipika.17u252@viit.ac.in_dev(London)
Code
Parameters
Runtime
Endpoints
Connected Triggers
Enclosing Sequences
Logs
Code
Node.js 10
Edit mode - press ESC to exit
Invoke with parameters
Invoke
/**
* @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}
*//**
* 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.
*/
const assert = require('assert');
const DiscoveryV1 = require('watson-developer-cloud/discovery/v1');
screenrec
13:17
25-05-2020


IBM Cloud Functions - Actions
IBM Watson Assistant
IBM Watson Discovery - Collectio
cloud.ibm.com/functions/details/action/dipika.17u252%2540viit.ac.in_dev//discovery-function/parameters
Paused
IBM Cloud
Catalog
Docs
Support
Manage
Dipika Aade's A...
Functions / Actions / discovery-function
discovery-function
Web Action
Namespace: dipika.17u252@viit.ac.in_dev(London)
Code
Parameters
Runtime
Endpoints
Connected Triggers
Enclosing Sequences
Logs
Parameters
Add Parameter
Parameter Name
Parameter Value
url
"https://api.eu-gb.discovery.watson.cloud.ibm.com/instance
environment_id
"d77a6b71-a96f-4424-ae0f-737290479c73"
collection_id
"a7059932-413a-41d7-9f69-5492049890e7"
iam_apikey
"VzpoTBXdjqJY519nO66SuToAv5rbWMpsfAycqgk6jl2f"
https://cloud.ibm.com/functions/details/action/dipika.17u252%2540viit.ac.in_dev/...
screenrec
13:18
25-05-2020

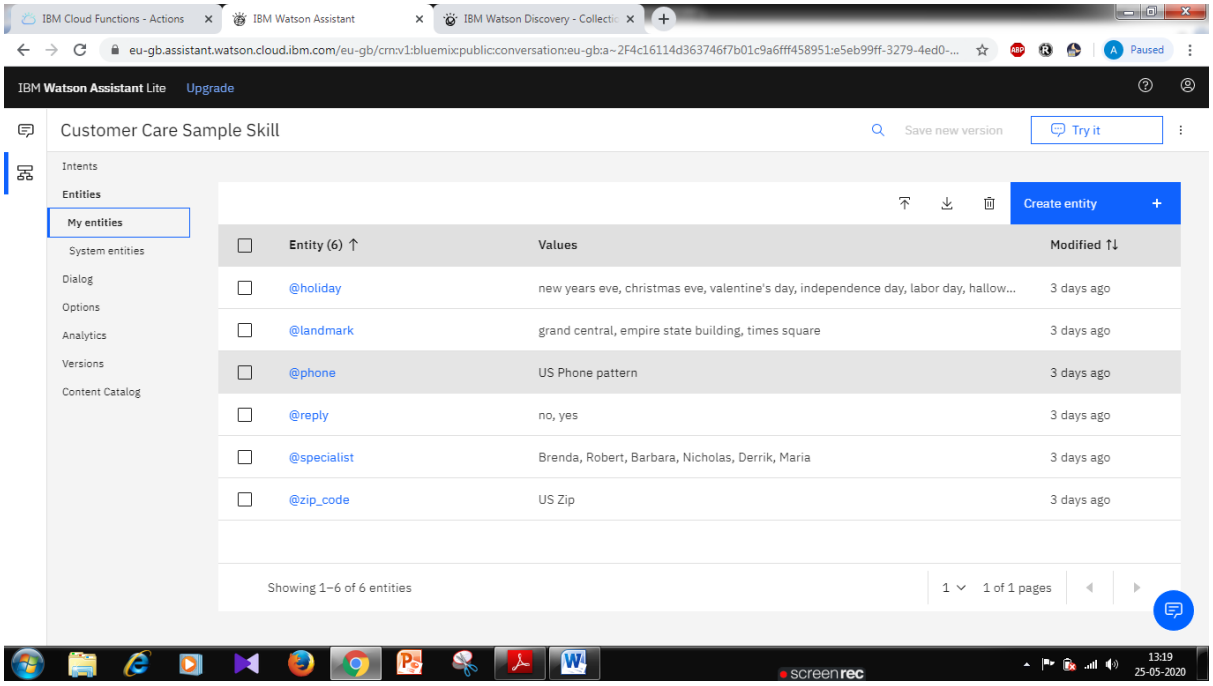


Next, we have to make the Watson assistant and use the sample customer care skill for convenience. We can add intent related to product information and the related entities and dialog flow.

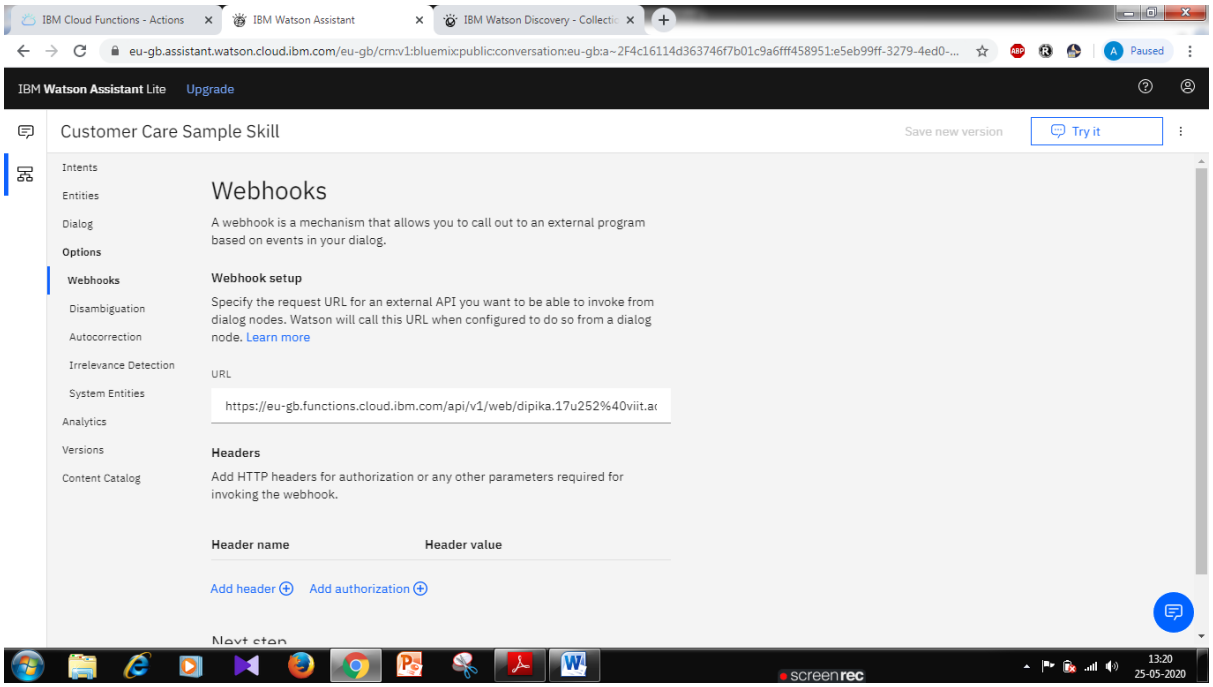
Intents- These are the categories which we mention or we expect the user input to be, for example: Greetings can be an intent and in it we can have examples as Good Morning, Good Evening and all.

Entities- These are used to mention the usual typos of the user and the synonyms like some people write the good morning as gm, good morning, gud morning, so we can cover all these also instead of returning a message to rephrase.

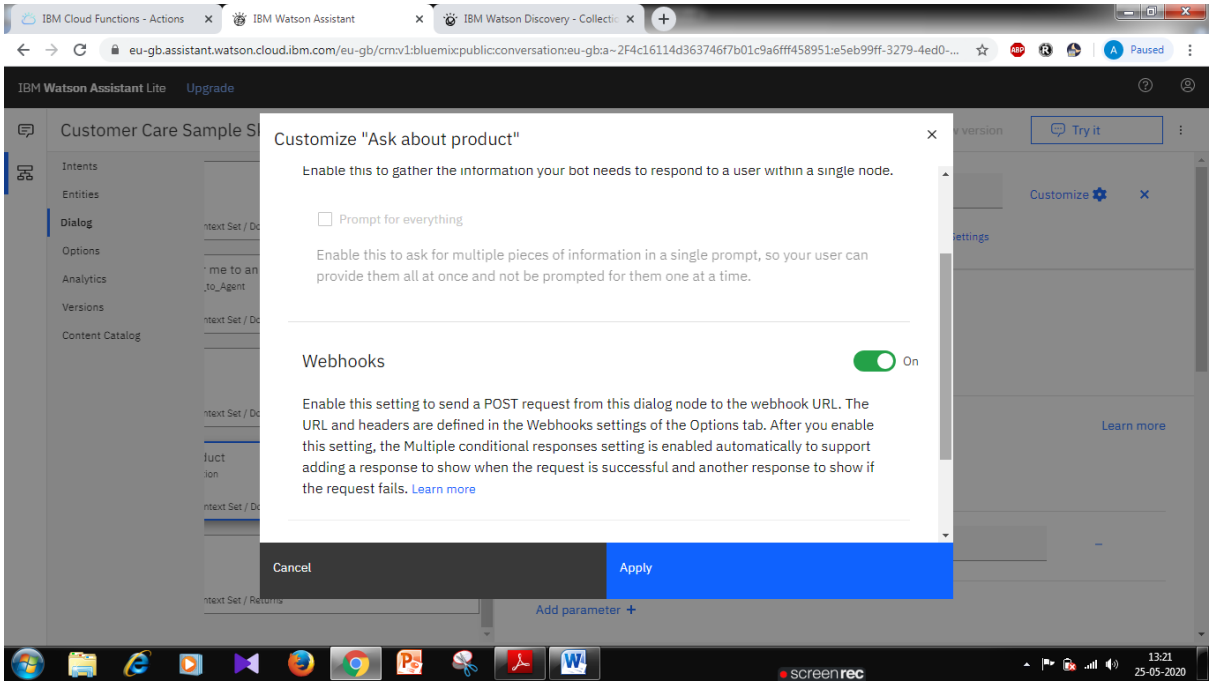
Dialog- Here we mention the outputs to be given, these can be static as well as dynamic.



Next, we have to mention the URL that we got earlier.

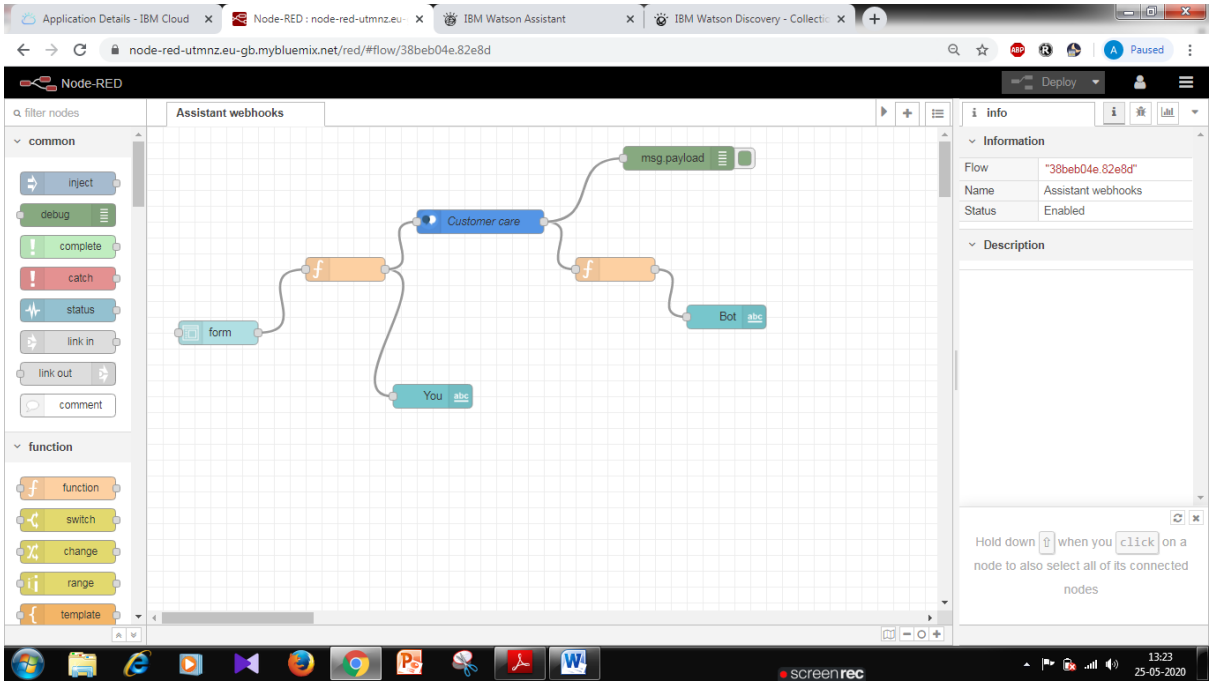


We have to enable the webhooks which enables our dialog to send a POST request to the webhook URL.

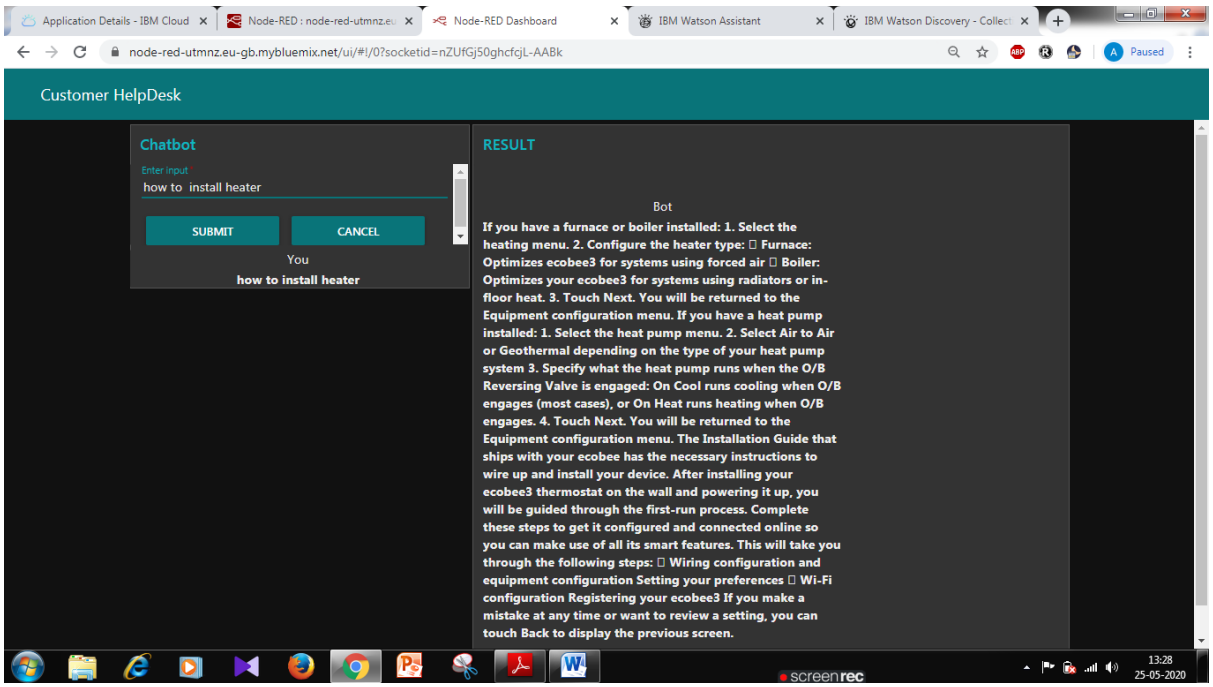


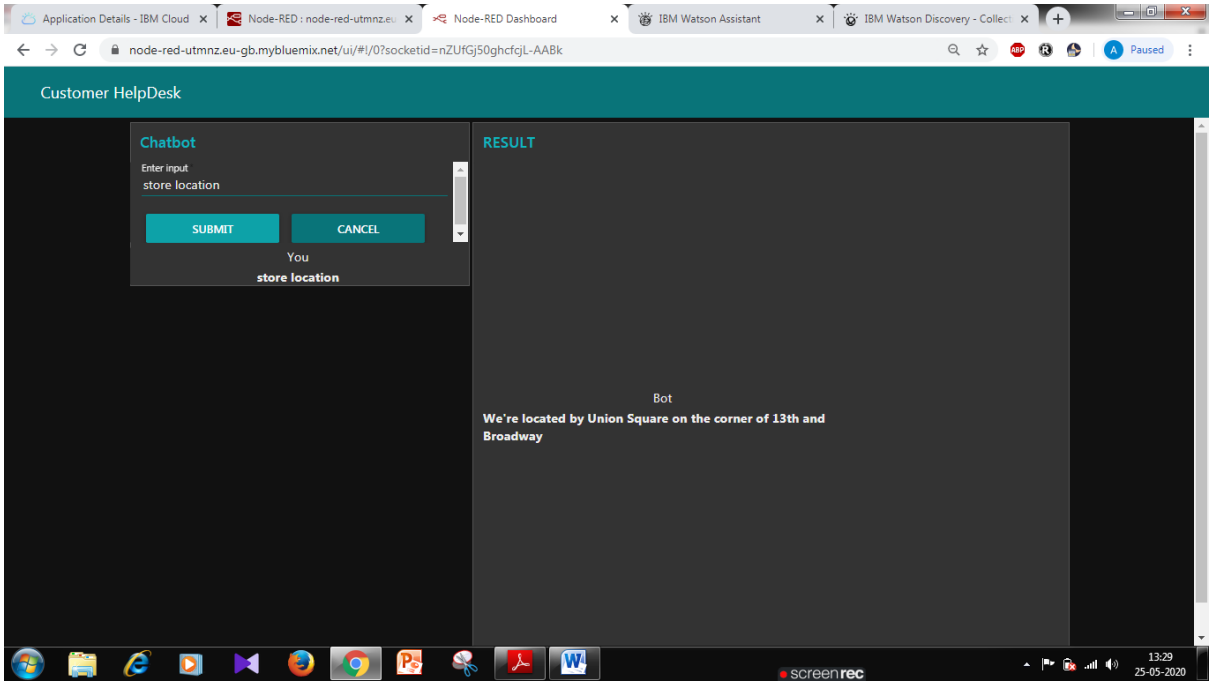
After this we have to make the node-red flow, and link everything. We will get a UI from the node. The roles of different nodes can be understood by the references mentioned in in the end. The final flow will look like as shown below.
The UI

5. Flowchart


Application Details - IBM Cloud
Node-RED : node-red-utmnz.eu
IBM Watson Assistant
IBM Watson Discovery - Collecti
node-red-utmnz.eu-gb.mybluemix.net/red/#flow/38beb04e.82e8d
Paused
Node-RED
Deploy
filter nodes
Assistant webhooks
common
inject
debug
complete
catch
status
link in
link out
comment
function
function
switch
change
range
template
msg.payload
Customer care
form
Bot
You
info
Information
Flow
"38beb04e.82e8d"
Name
Assistant webhooks
Status
Enabled
Description
Hold down ⇧ when you click on a node to also select all of its connected nodes
screenrec
13:23
25-05-2020

6. Result


Application Details - IBM Cloud
Node-RED : node-red-utmnz.eu
Node-RED Dashboard
IBM Watson Assistant
IBM Watson Discovery - Collect
node-red-utmnz.eu-gb.mybluemix.net/ui/#!/0?socketid=nZUfGj50ghcfcjL-AABk
Paused
Customer HelpDesk
Chatbot
Enter input *
how to install heater
SUBMIT
CANCEL
You
how to install heater
RESULT
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. If you have a heat pump installed: 1. Select the heat pump menu. 2. Select Air to Air or Geothermal depending on the type of your heat pump system 3. Specify what the heat pump runs when the O/B Reversing Valve is engaged: On Cool runs cooling when O/B engages (most cases), or On Heat runs heating when O/B engages. 4. Touch Next. You will be returned to the Equipment configuration menu. The Installation Guide that ships with your ecobee has the necessary instructions to wire up and install your device. After installing your ecobee3 thermostat on the wall and powering it up, you will be guided through the first-run process. Complete these steps to get it configured and connected online so you can make use of all its smart features. This will take you through the following steps: □ Wiring configuration and equipment configuration Setting your preferences □ Wi-Fi configuration Registering your ecobee3 If you make a mistake at any time or want to review a setting, you can touch Back to display the previous screen.
screenrec
13:28
25-05-2020



7. Advantages and Disadvantages

Advantages:

- Companies can use these to decrease the work flow to the representatives.
- Reduce the number of reps.
- Cost Efficient.
- Decrease in the number of calls diverted to representatives.
- Less work load on employees.

Disadvantages:

- The discovery returns wrong results when not properly configured.
- Giving same answer for different sentiments.
- Sometimes is unable to connect the customer sentiments and intents.

8. Application

• It can be deployed in popular social media applications like Facebook, Slack and Telegram.

• Chatbot can be deployed at any website to clear the basic doubts of the customer.

9. Conclusion

By following the above-mentioned steps, we can create a basic chatbot which can help us to answer the basic questions of the customer or user related to location of the office, working hours and the information about the product. We successfully create the intelligent helpdesk smart chatbot using Watson Assistant, Watson Cloud Function, Watson Discovery and Node-Red.

10. Future Scope

We can import the pre-built node-red flow and can improve our UI, moreover we can make a data base and use it to show the recent chats to the customer. We can also improve the results of discovery by enriching it with more fields and doing the Smart Data Annotation more accurately. We can get the premium version to increase the scope of our chatbot in terms of the calla and requests.

We can also include Watson text to audio and Speech to text services to access the chatbot handsfree. These are few of the future scopes which are possible.

11. Appendix

11.1 Code:

Cloud Function:

```
/**
*
* @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': '2020-05-09'
});
}
else {
discovery = new DiscoveryV1({ 'username': params.username, 'password':
params.password, 'url': params.url,
'version': '2020-05-11'
});
}
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);
});
});
}
```

11.2 References

- • https://www.ibm.com/cloud/architecture/tutorials/cognitive_discovery
- • https://cloud.ibm.com/docs/assistant?topic=assistant-getting-started
- • https://developer.ibm.com/recipes/tutorials/how-to-create-a-watson-chatbot-on-nodered/
- • http://www.iotgyan.com/learning-resource/integration-of-watson-assistant-to-node-red
- • https://github.com/IBM/watson-discovery-sdu-with-assistant
- • https://www.youtube.com/watch?v=Jpr3wVH3FVA