Project Scope

Project Scope :

● 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
● Build a web application with integration to all these services &
deploy the same on IBM Cloud Platform

Schedule : 30 days
Project Team : Rishabh Dixit
Project Deliverables : Intelligent Customer Care Chatbot with Smart Document Understanding.

Project Report

Project Name:Intelligent Customer Help Desk with Smart Document Understanding

Date: May 04,2020

Project Manager:Rishabh Dixit
Email ID:rishabhdixitjp@gmail.com

Category:Artificial Intelligence

Contents:

11.1. Source codes

1.Introduction

1.1 Overview

We will be able to write an application that leverages multiple Watson AI services like Discovery, Assistant, Cloud Function and Node Red. By the end of the project we will learn best practices of combining Watson Services, and how they can build interactive retrieval system with discovery + assistant.

- Project Requirements: Python, IBM Cloud, IBM Watson
- Functional Requirements: IBM Cloud
- Technical Requirements: Python, Watson AI, ML
- Software Requirements: Watson Assistant, Watson Discovery
- Project Deliverables: Smartinternz Internship • Project Duration: 1 Month

1.2 Purpose

The chatbot usually can answer just simple questions, such as store locations and hours, directions and maybe even making appointments. When any user input (question) falls out of the scope of the predetermined question set of chatbot, it typically tells us to rephrase our sentence or mention that this question is not valid. In this project we will be emphasizing to reduce the workload from the representative by supervising our chatbot in such a way that it is able to answer much of the questions (in particular the product information). We can return the relevant sections from the user guide which can help us to reduce the number of clients of every representative at a call centre. We will be transferring the call to representatives in particular cases only. To take this a step further, the project shall use the Smart Document Understanding provided by the Watson Discovery to train on the text, which is a modified version of Natural Language Processing.

Scope of Work:

• Create a customer care dialog skill in Watson Assistant.
• Use smart document understanding to build an enhanced Watson Discovery Collection.
• Create an IBM Cloud Function web action that allows Watson Assistant to post queries to Watson Discovery.
• Build a web application with integration of all the services and deploy the same on IBM Cloud Platform.

2. Literature Survey

2.1 Existing Problems

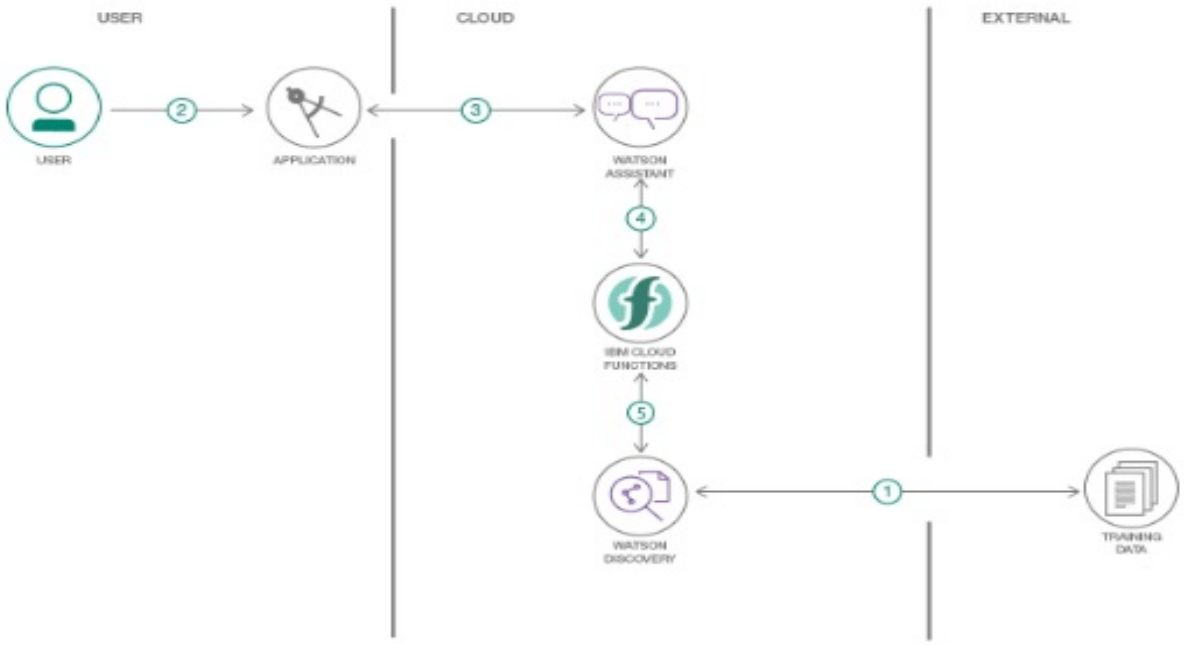
Generally chatbot are loaded with a certain set of questions that is more like if and else flow, the question or the user input which lies out of the scope of the chatbot is not answered and rather a message like “Try again after rephrasing” & “I am unable to understand, Please Rephrase” are displayed and it directs the user to the customer agent or the representative but an efficient chatbot should reduce the traffic reaching to the representatives, So to achieve this we include a Smart chatbot so that it can answer the queries of the customer.

2.2 Proposed solution

For the above-mentioned problem, we have to put a virtual agent in chatbot so it can understand the queries that are posted by customers. The virtual agent should be trained from some insight based on company backgrounds, working hours, store locations and product related information. In this project I used Watson Discovery to achieve the above solution.

3. Theoritical analysis

3.1 Block/Flow diagram



1. The document is annotated using Watson Discovery SDU.
2. The user interacts with the backend server via the app UI. The frontend app UI is a chatbot 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 the product operation question, a search query is passed to a predefined IBM Cloud Function action.
5. Cloud function action will query the Watson Discovery service and return the results.

3.2 Hardware/Software Designing

- Create IBM Cloud services.
- Configure Watson Discovery
- Create IBM Cloud Function action
- Configure Watson Assistant
- Create flow and configure node
- Deploy and run node red app

4. Experimental Investigation

1) Create IBM Cloud Services

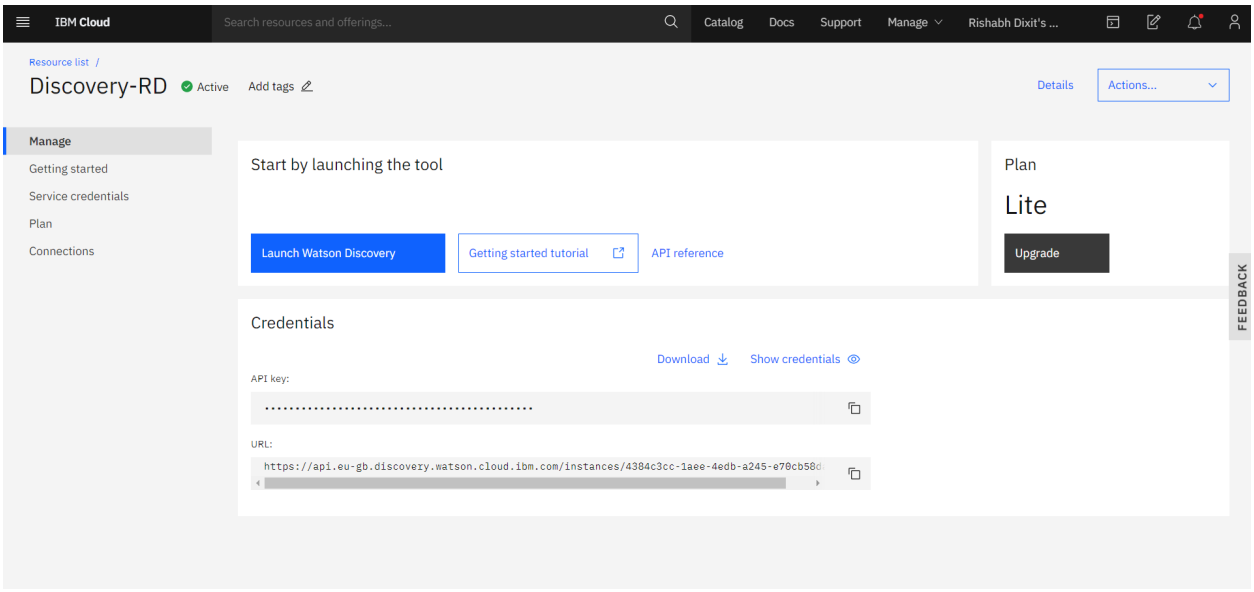
a. Watson Discovery
b. Watson Assistant
c. Node Red

2) Configure Watson Discovery

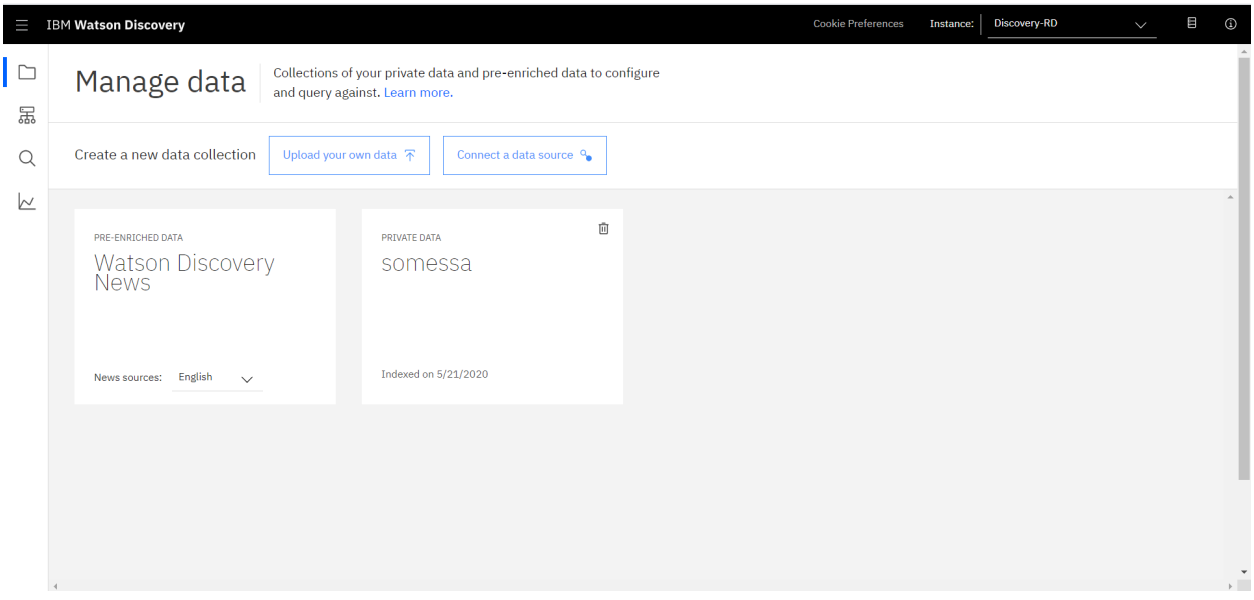
After creating and launching the discovery from the Catalog, Import the document on which on which we need to train the discovery service. We have selected the ecobee3 user guide located in the data directory of our local repository. The Ecobee3 is a popular residential thermostat that has a WIFI interface 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:

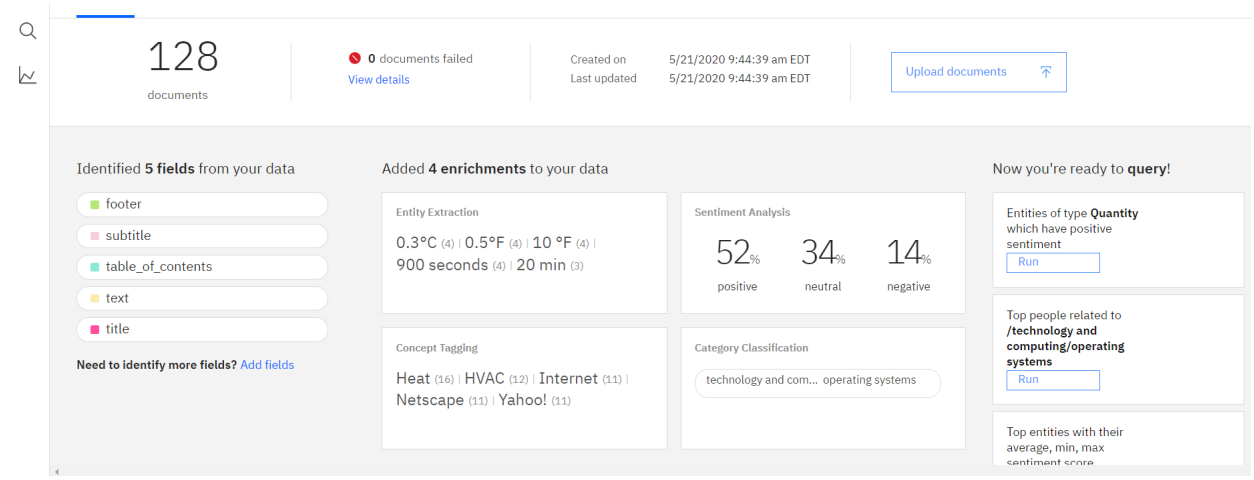
• After creating the discovery from the catalog, we will be redirected to this base page of discovery where the name of the discovery along with its API Key and URL are mentioned. These credentials will be used in further steps.



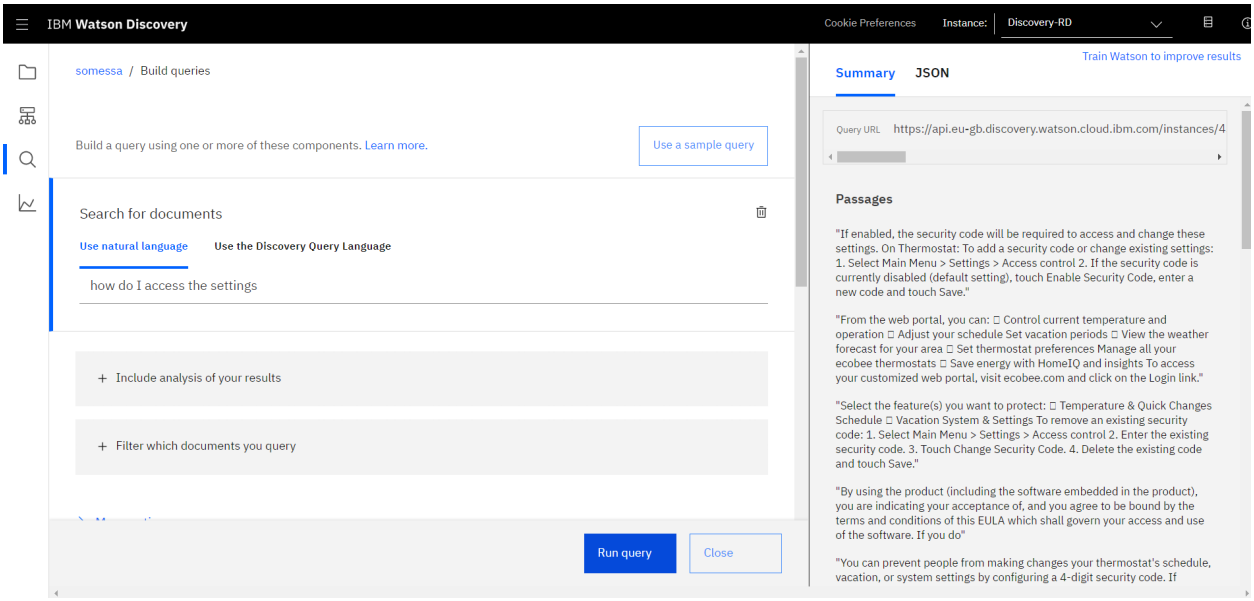
Click on the Launch Watson Discovery [1] to launch the discovery. • Now in the next step we have to upload the data by clicking, upload your data. Here we have already uploaded the data as manual.



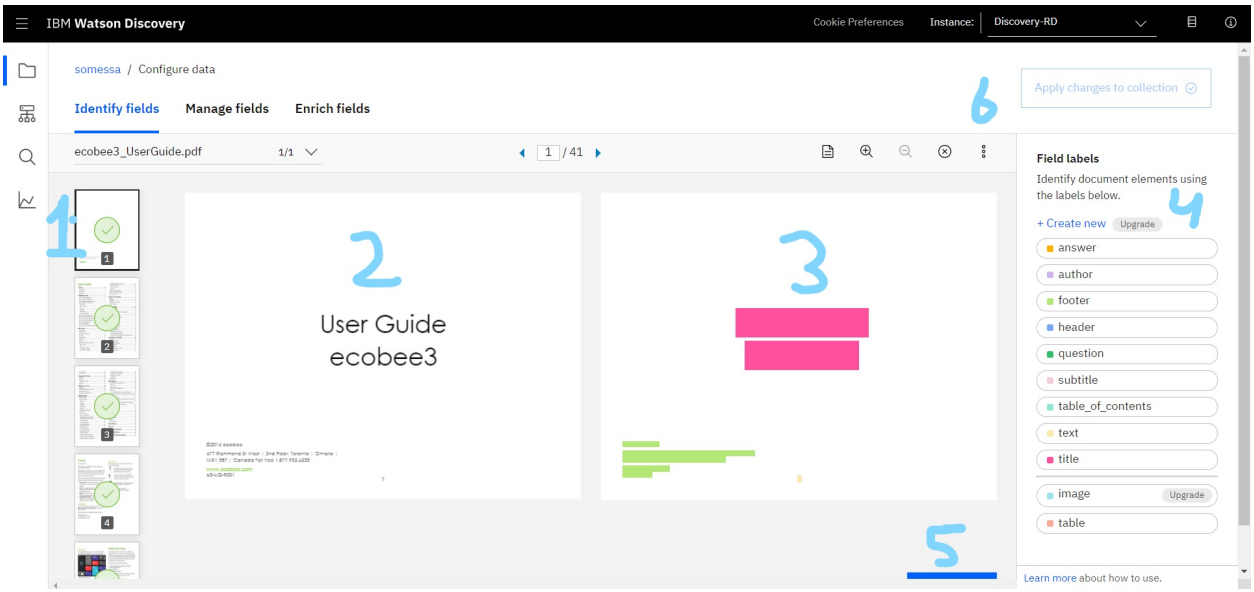
• After uploading the document we will see the page like this, we can ignore the warnings section as it is due to the normalization process and is of no worry. Now click on the build your own query [1] , so that we can have an idea how the results have significantly improved after configuring the data.



• 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:



The aim is to annotate all the pages of the document, so that discovery can learn what text is important and what text can be ignored.

[1] is the list of pages we have in the document. In the lite plan we are unable to upload a document more than 50,000 words and any document with more than that will be trimmed to this range.

[2] is the current page being annotated.

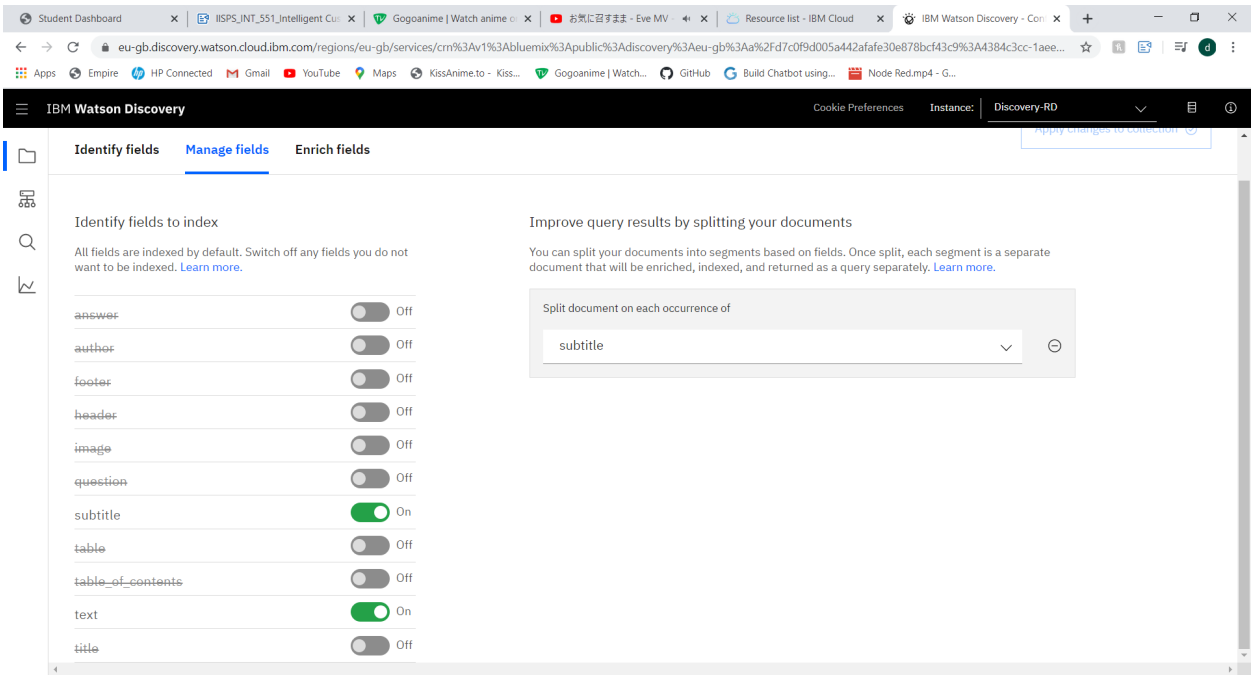
[3] is the page in which we have to provide a label to each text, for example as shown above we have the Title and text, at the bottom we have the footer.

[4] is the list of labels we can provide in which the image label is not available in the lite plan. We have to click the submit page button

[5] after annotating each page individually, after few pages the discovery will automatically give the label to the remaining pages.

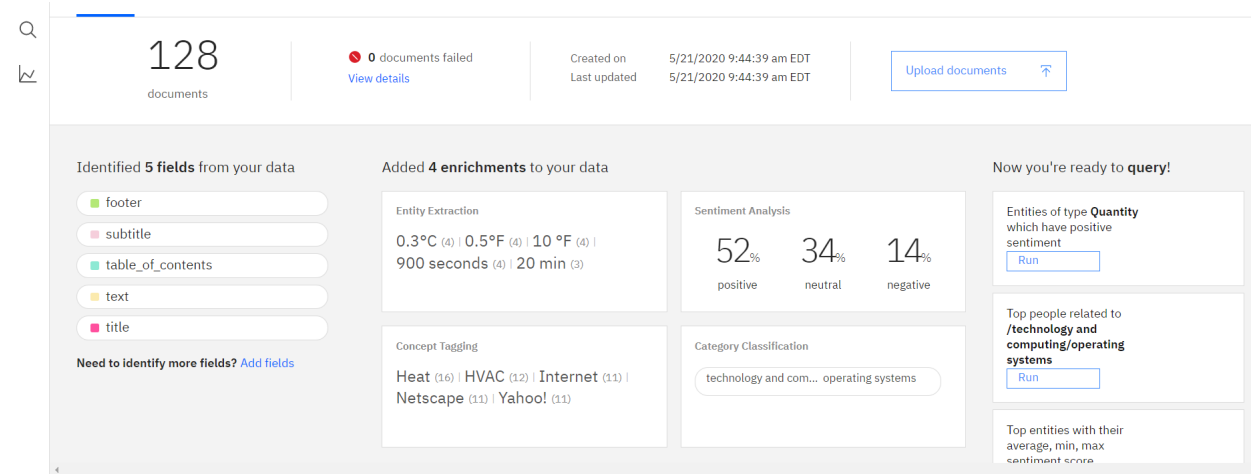
Click [6] after completing the annotation of the document.

• 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.

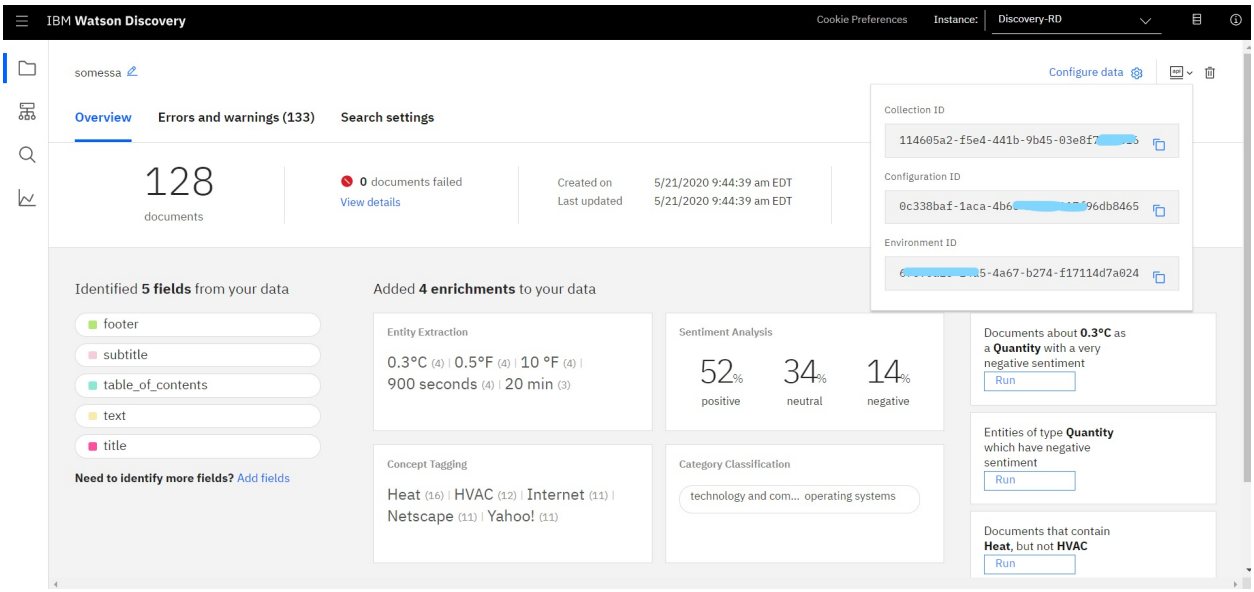


After doing everything as mentioned above, we have to click on Apply changes to Collection.

• It will take some time to process and after that we will have multiple documents as shown below, the document we uploaded earlier is segmented in 128 documents as shown below.

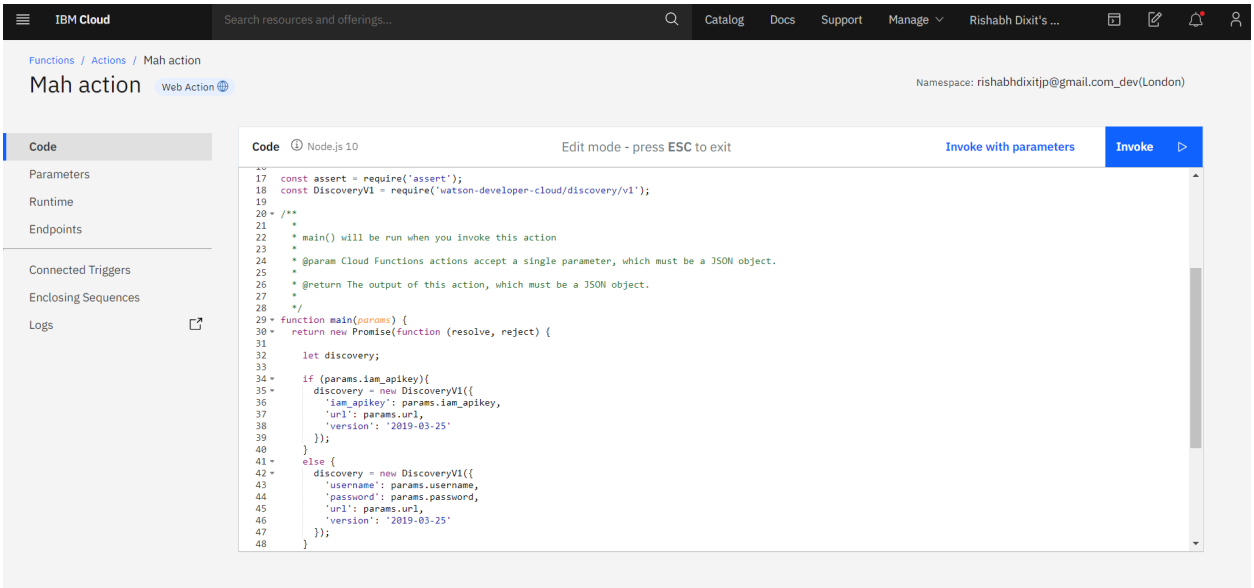


• Next, we have to store the credentials of Discovery which can be viewed as shown below:

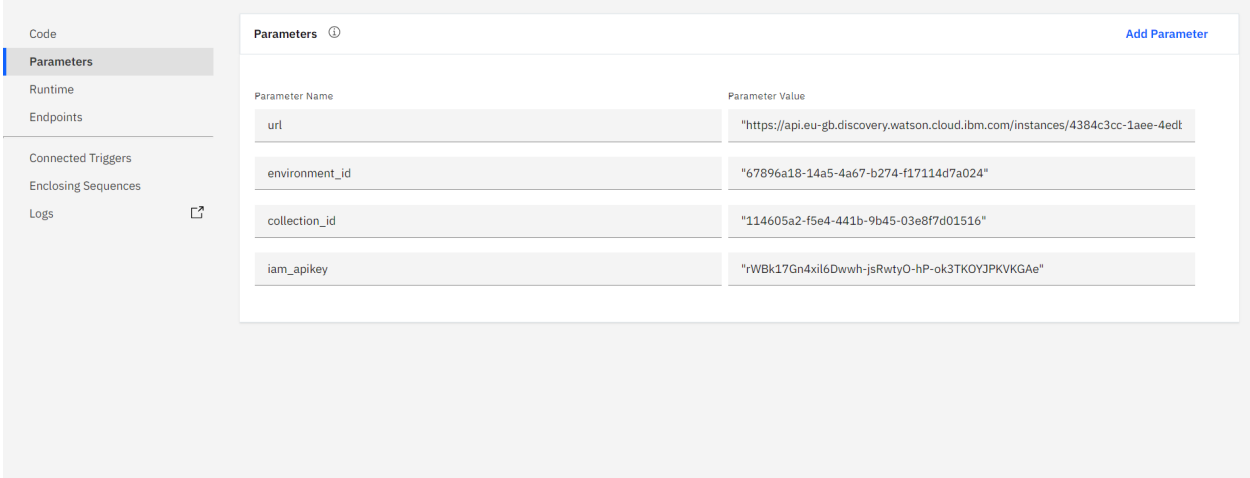


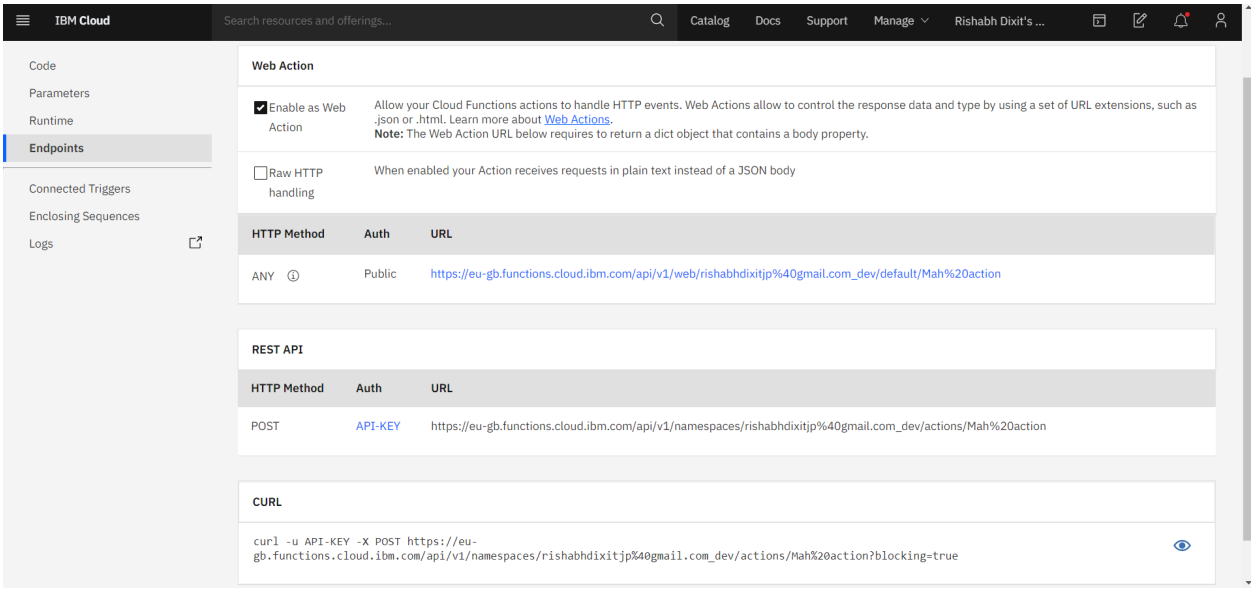
• We already have the API key and the URL.

• Next, we have to create the IBM Cloud Function Action: 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.

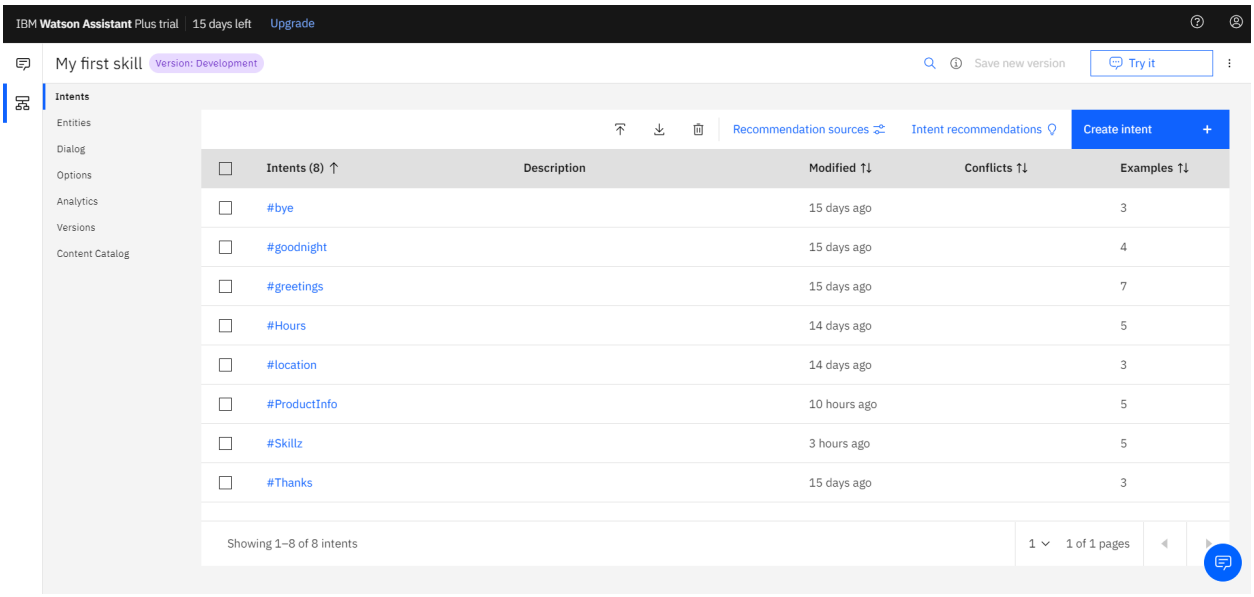


We can make the parameters as per the code and paste the parameter value from the discovery credentials. After that we have to click on the endpoint and enable the web action which will generate a public URL and it will be further used.

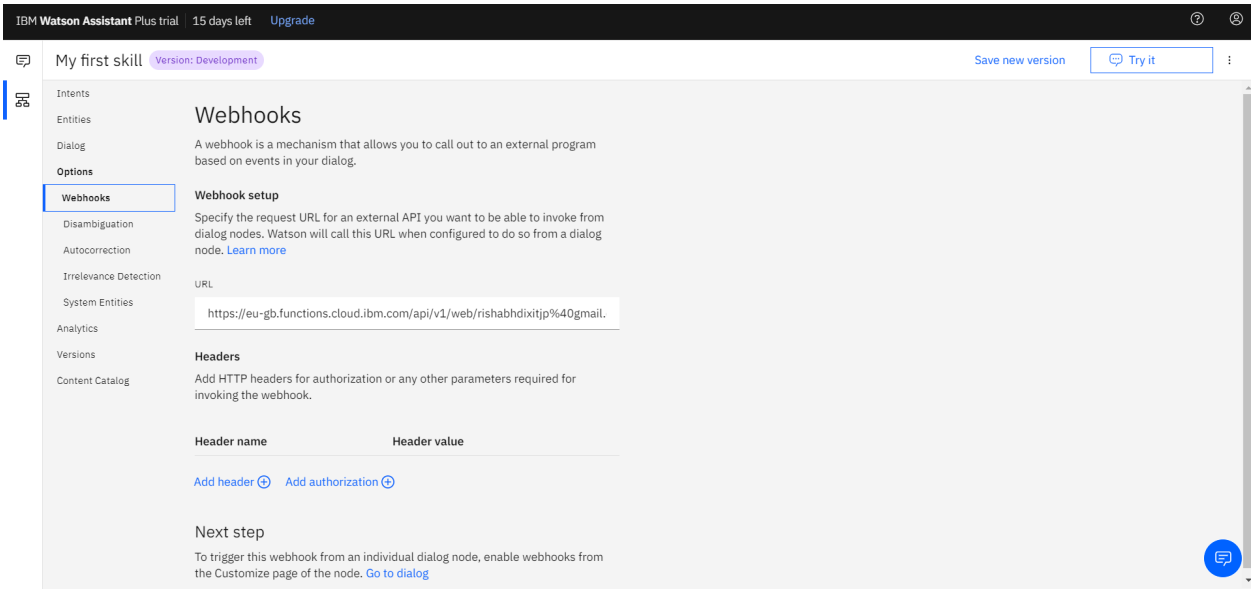




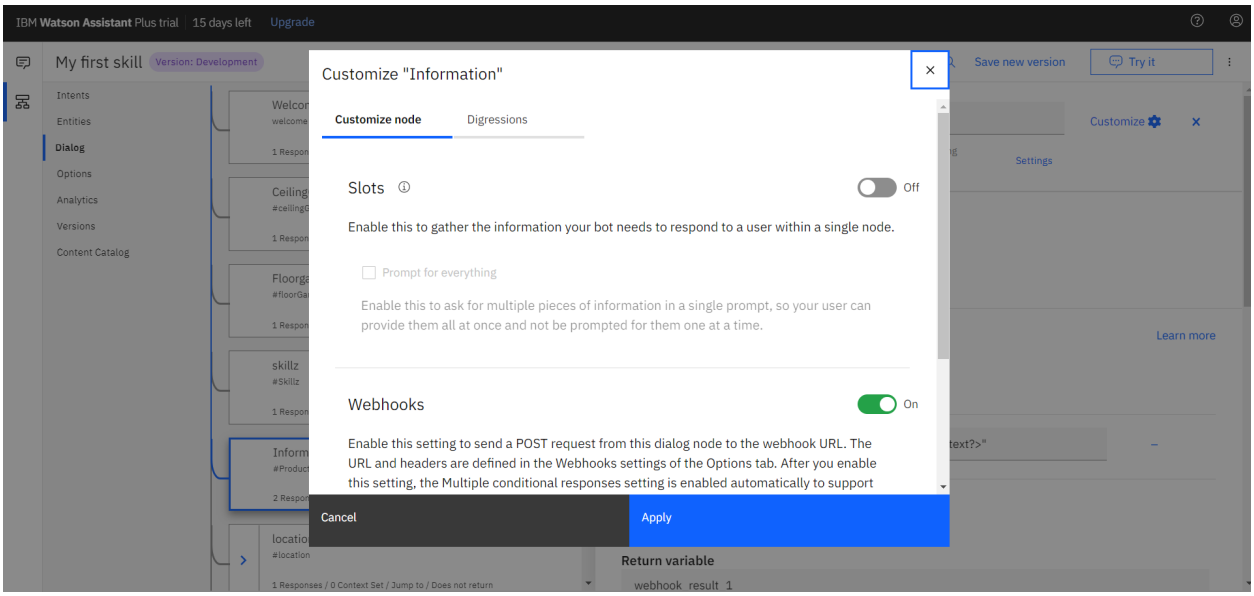
We can make the parameters as per the code and paste the parameter value from the discovery credentials. After that we have to click on the endpoint and enable the web action which will generate a public URL and it will be further used. • Next, we have to make the Watson assistant and use the sample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



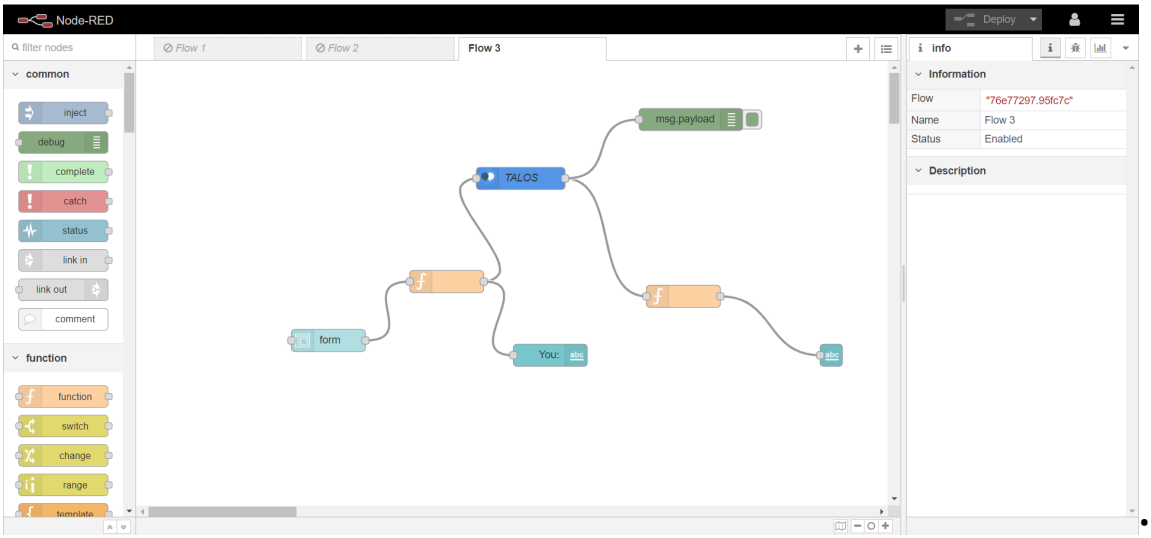
• Next, we have to mention the URL that we got earlier.



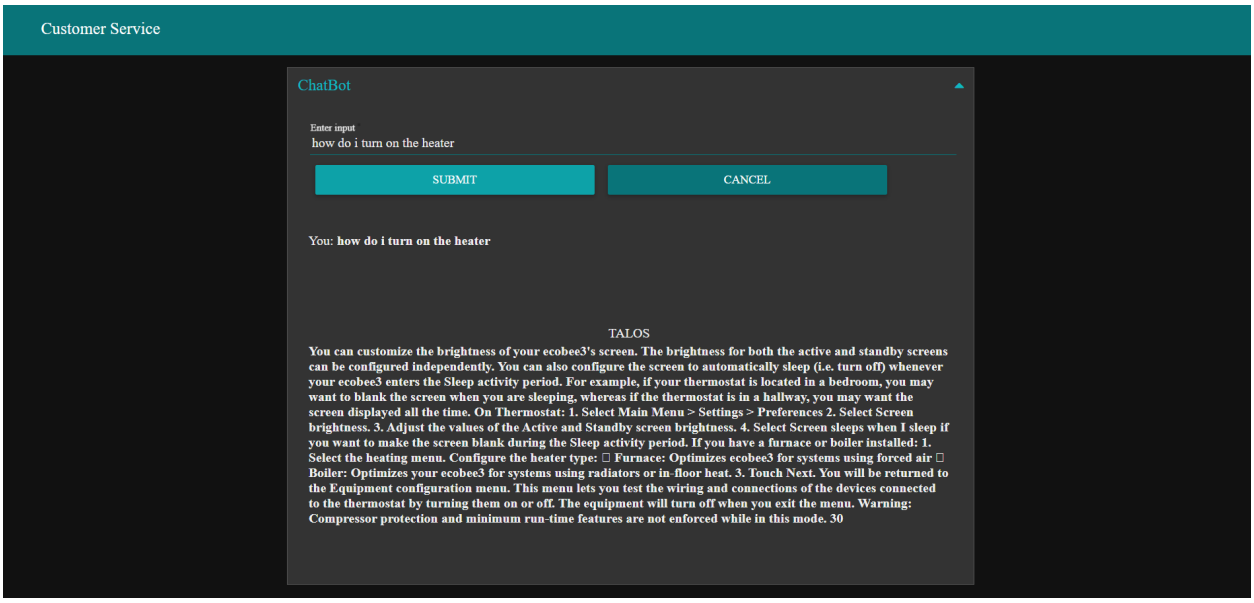
We have to enable the webhooks which enables our dialog to send a POST request to the webhook URL.



• 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.



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 we have is the basic one but can be improved by writing the HTML code in the template node. We can vary the background colour also from the node-red.


Customer Service
ChatBot
Enter input
how do i turn on the heater
SUBMIT
CANCEL
You: how do i turn on the heater
TALOS
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. 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

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:

- Sometimes the chatbot misleads the customers.
- 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': '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);
    });
  });
}
```