**PROJECT REPORT**

**TOPIC:** Intelligent Customer Help Desk with Smart Documentation Understanding

**Name:** Nitin Bhoopal

**Email ID:** nitinbhoopal@gmail.com

**Category:** Artificial Intelligence

Internship at smartinternz.com@2020

1. **Introduction**
   1. **Overview:**

We shall build an application that utilizes various Watson AI services like Discovery, Assistant, Cloud Function and Node Red. We shall use Watson Assistant to make the chatbot, Discovery to analyse the document, Cloud functions to connect the chatbot to the document, and finally Node Red to integrate them all.

• Project Requirements: Python, IBM Cloud, IBM Watson

• Functional Requirements: IBM Cloud

• Technical Requirements: Python, Watson AI, ML

• Software Requirements: Watson Assistant, Watson Discovery ,Node Red

• Project Deliverables: Smartinternz Internship

• Project Duration: 1 Month

* 1. **Purpose**

The project's goal is to make a customer care chatbot which can answer simple questions such as location of the store, directions to the store, store timings or even book appointments. Furthermore, if there are some queries regarding the product and the information is available in the user's manual then the chatbot can extract the result from the manual as well.

The main objective of this chatbot is to support and scale business teams in their relations with customers and also decrease the user traffic being redirected to customer care.

1. **Literature Survey**

**2.1 Existing Problem**

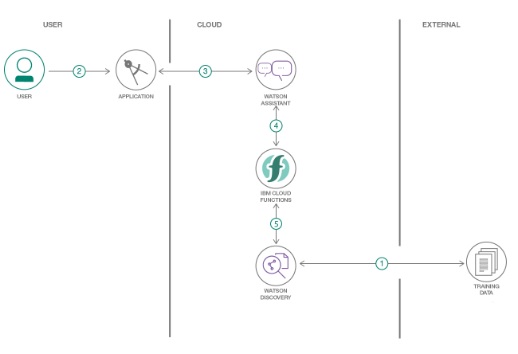
The issue with the old system was that when replacing a customer care executive, the recruit had to be trained, and would hence, require significant increases in capital and time.

**2.2 Proposed Solution**

A solution to the existing problem was to implement chatbots. Chatbots required a one-time investment and consequently lead to drastic reductions in expenditure as the overhaul of retraining was no longer necessary. The virtual agent should be trained on company rules, timings, store locations and product related information.

1. **Theoretical Analysis:**

**3.1. Block 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 a question outside its usual scope and product related, 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

1. **EXPERIMENTAL INVESTIGATIONS**
2. **Create IBM Cloud Services** 
   1. Watson Discovery
   2. Watson Assistant
   3. Node Red
3. **Configure Watson Discovery**

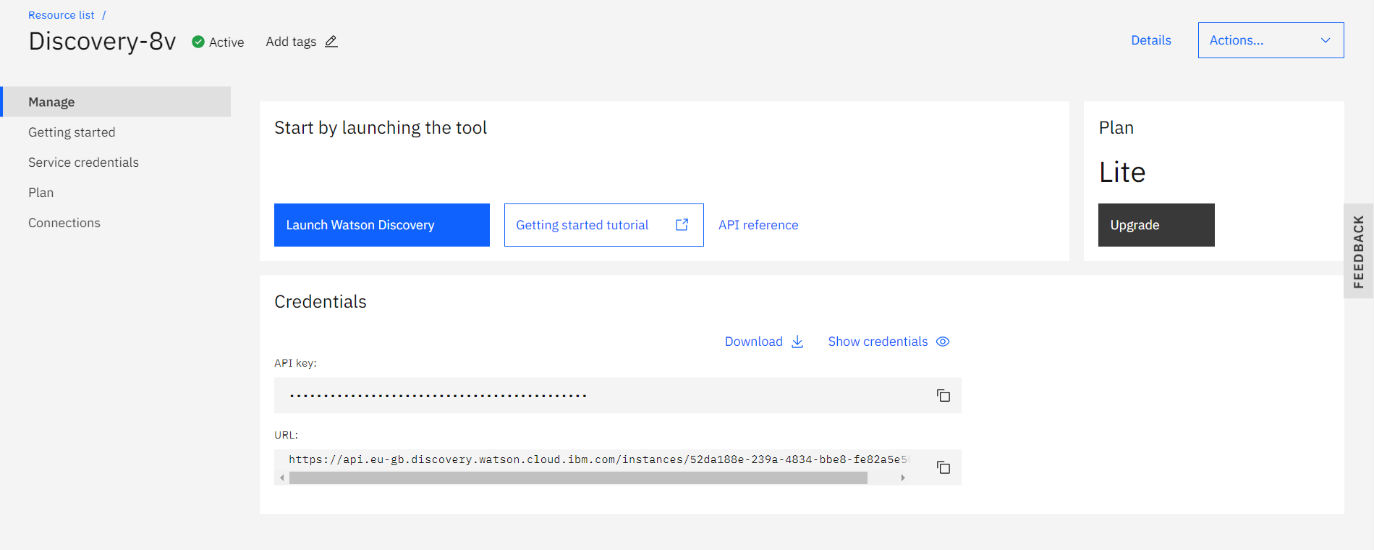
After creating and launching the discovery from the catalog, Import the document required for smart document understanding. I have selected the Samsung Refrigerator user manual for model DA68-02966A-05.

The result of the queries performed without configuring the data present in the document won’t be accurate. The results improve significantly after applying SDU (Smart Document Understanding). This is done by identifying key elements present in the document such as the title, subtitle, text, table, image, and footer.

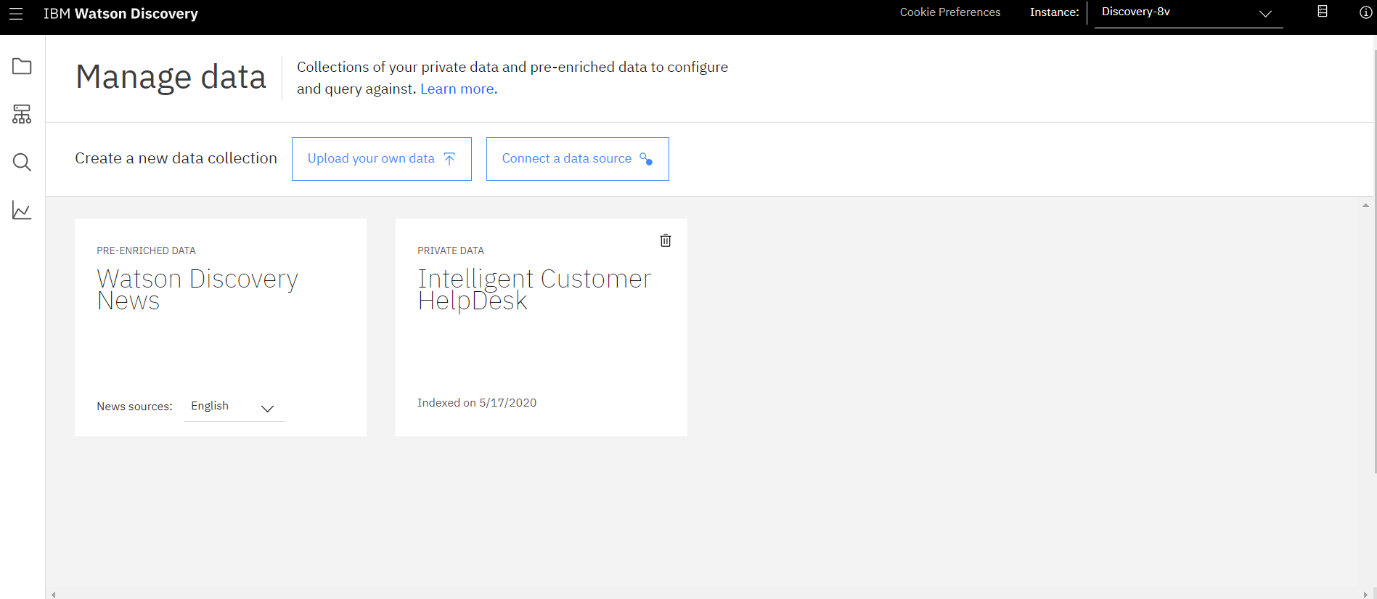
In the lite plan we are provided with limited content of IBM Watson, the labels help us in segmentation of the document which helps the discovery to understand the document better and provide better results. Watson Discovery runs Deep Learning algorithms on the document with the respective key elements considered to categorize the manual into useful and useless information. The document can then be split based on the specific field selected. 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.

Follow the below mentioned steps:

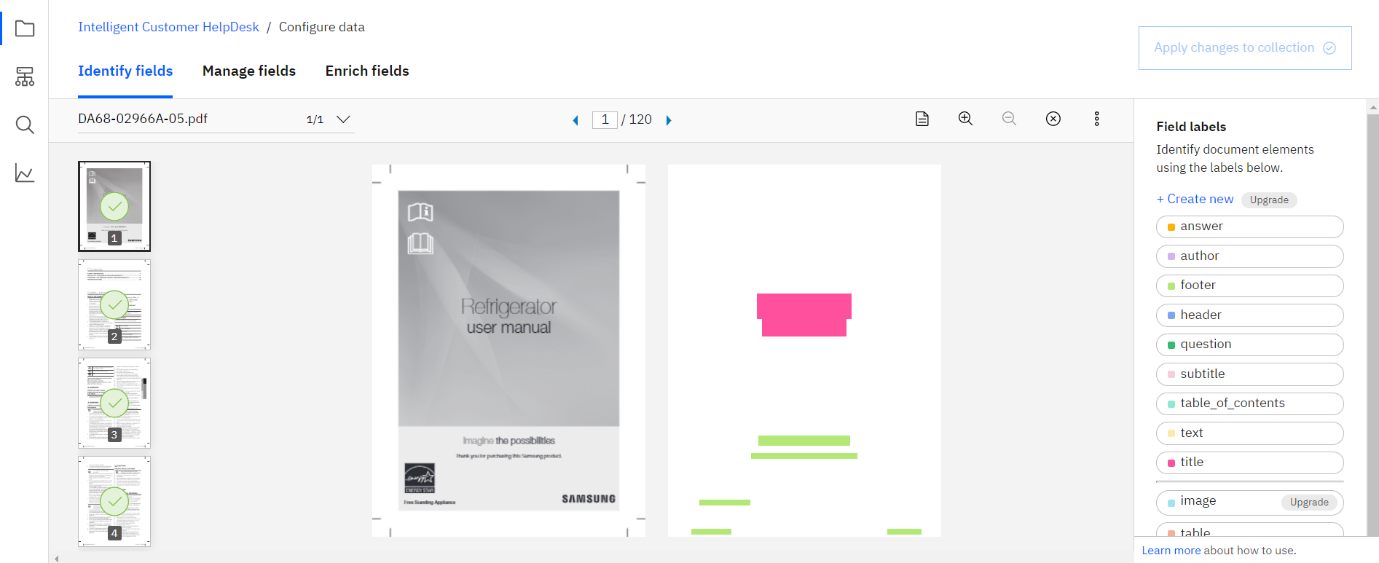
* After creating the discovery from the catalog, we will be redirected to this base page of discovery where the name of the discovery along with its API Key and URL are mentioned. These credentials will be used in further steps.



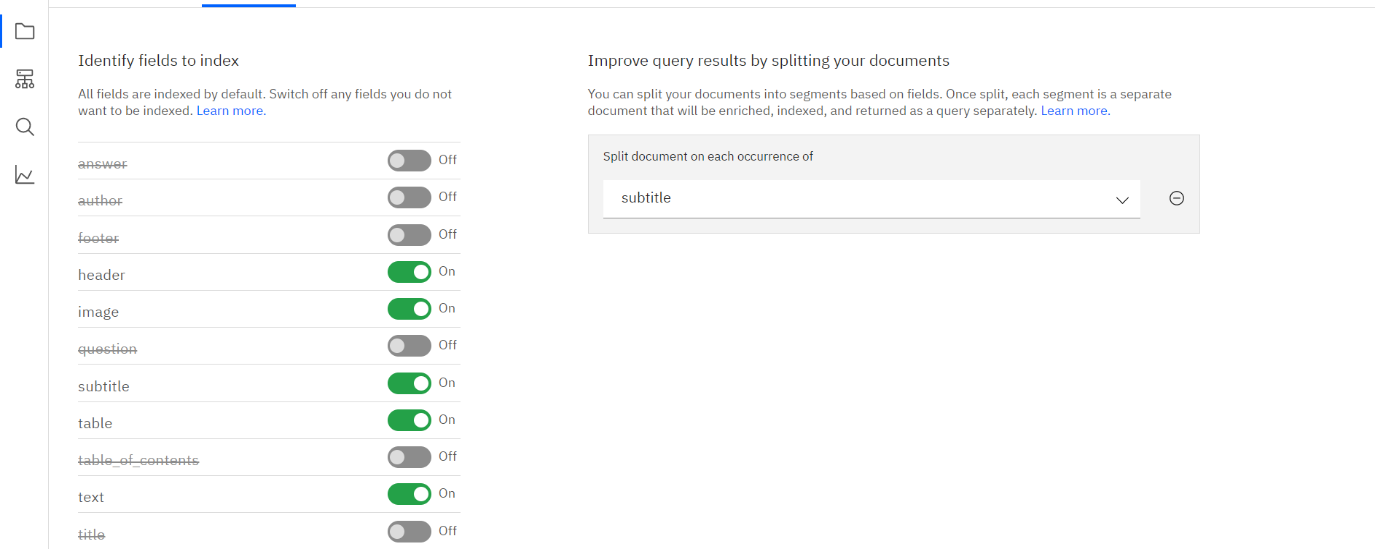
* Click on the Launch Watson Discovery to launch the discovery.
* Now in the next step we have to upload the data by clicking, upload your data. Here we have already uploaded the document.



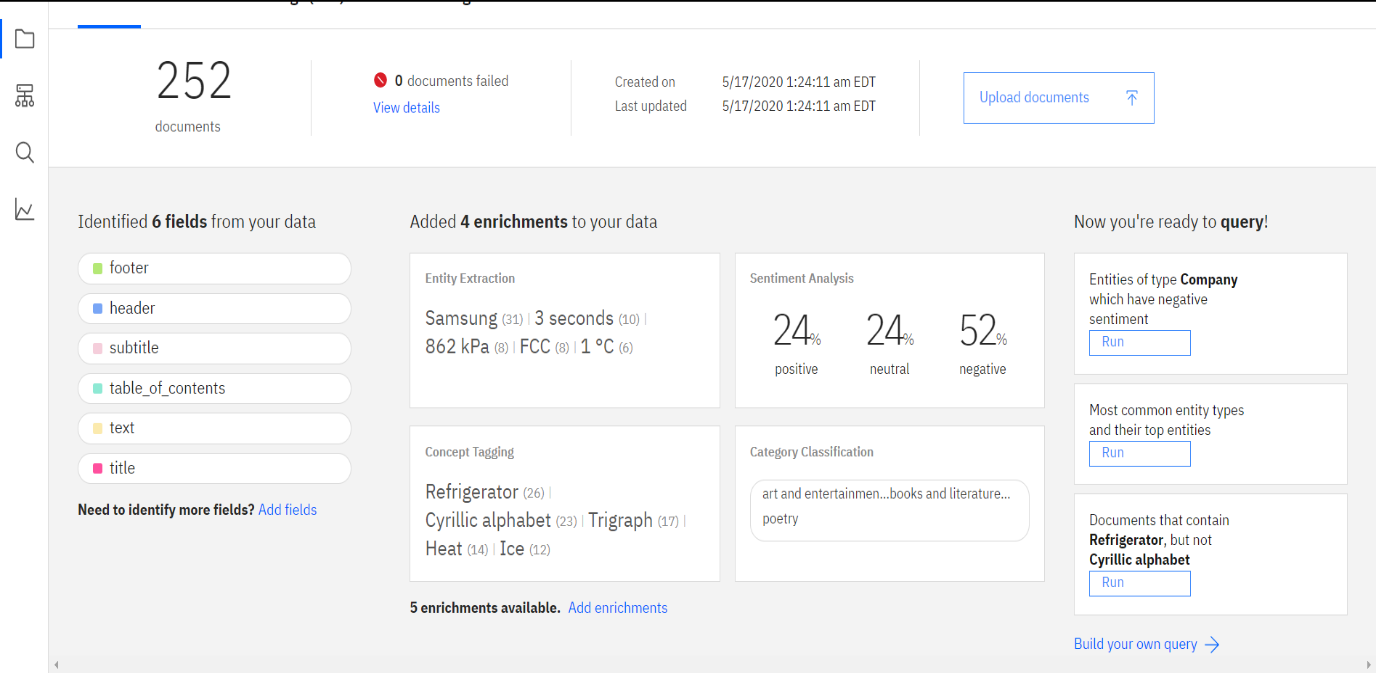
* After uploading the document, click on the build your own query and run an example. We will notice that the results are not very accurate and to improve this we need to configure the document by clicking on configure data. The next step is to annotate the document with SDU.
* Below is the layout of Identify fields tab of the SDU annotation panel:



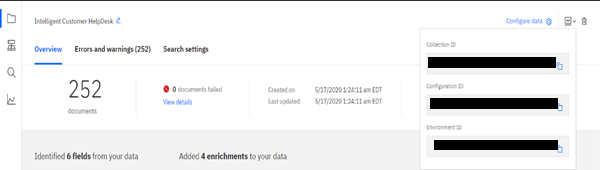
* The aim is to annotate most of the pages so that discovery can learn to distinguish between subfields.
* For further segmentation and making the sub documents, we have to manage the fields. Here we are provided with the option of identifying field to index i.e. important texts, given below are the field such as subtitle, title and text because they are the only labels which are useful for us. 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.



* Processing the document will take some time. On completion, you will see the document being split into multiple other documents. This is shown below with a total of 252 documents.

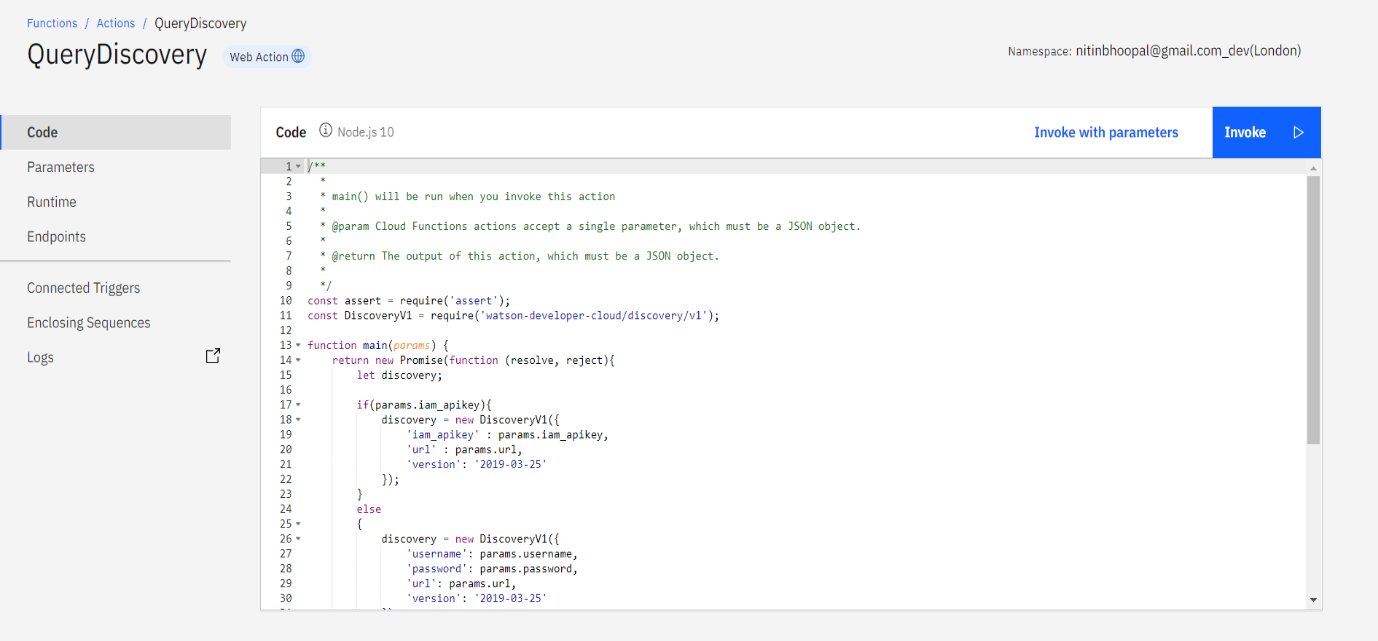


* Return to the query panel and try to rerun the query. We shall a drastic improvement in the answer. This is because of the correct labelling of the document.
* Next, we have to store the credentials of Discovery which can be viewed as shown below:

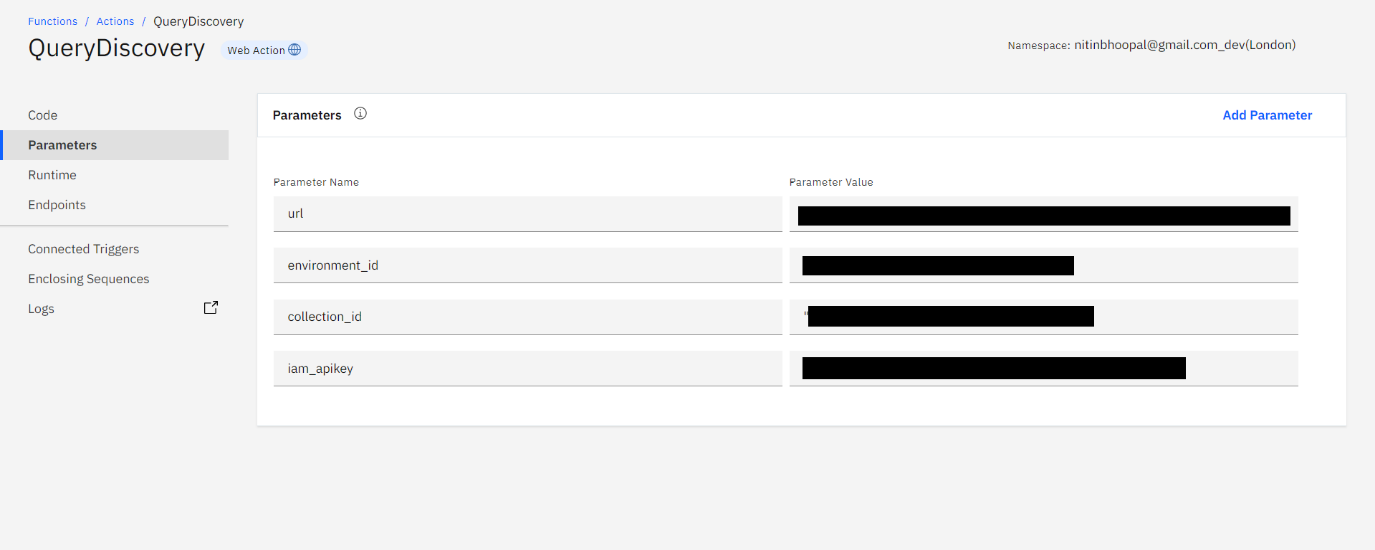


**3. Cloud Function**

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.



* The parameters have to be in accordance to the variables used in the code and the parameter values are the Watson discovery credentials. After that, click on the endpoints tab and enable web action which will generate a public URL and it will be further used.



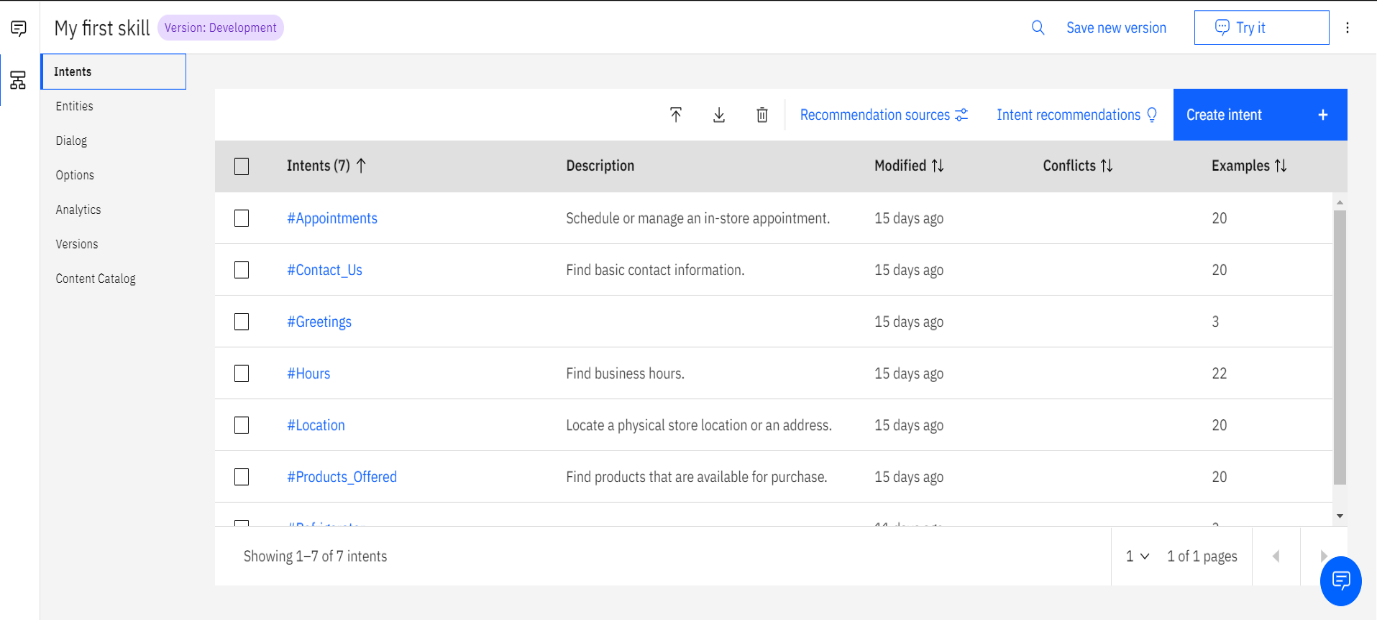
**4. Watson Assistant**

* Next, we have to configure Watson Assistant and use the sample customer care skill to avoid having to manually create the intents and the entities. We can add additional intents and entities as required.

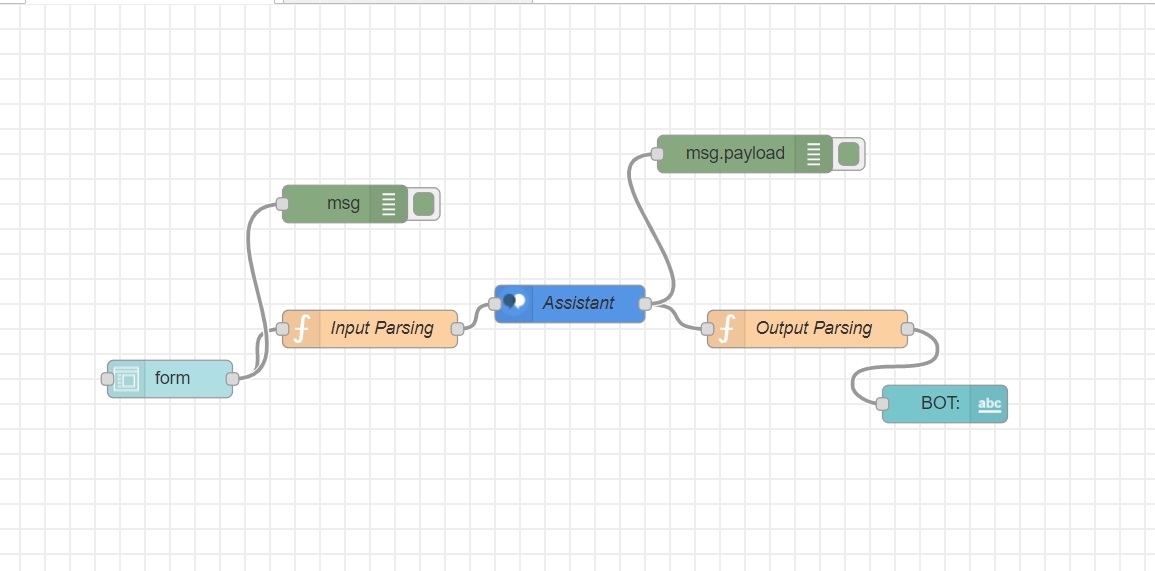
Intents- These are the broad categories used to identify and classify dialogs in a chatbot. These are what expect the user to input, for example: We could have an intent called Appointments, which has examples such as: Can I make an appointment?, Book an appointment and other similar phrases.

Entities- These are more specific data items. They usually represent the terms commonly used by users to refer to a certain action or topic and their synonyms.

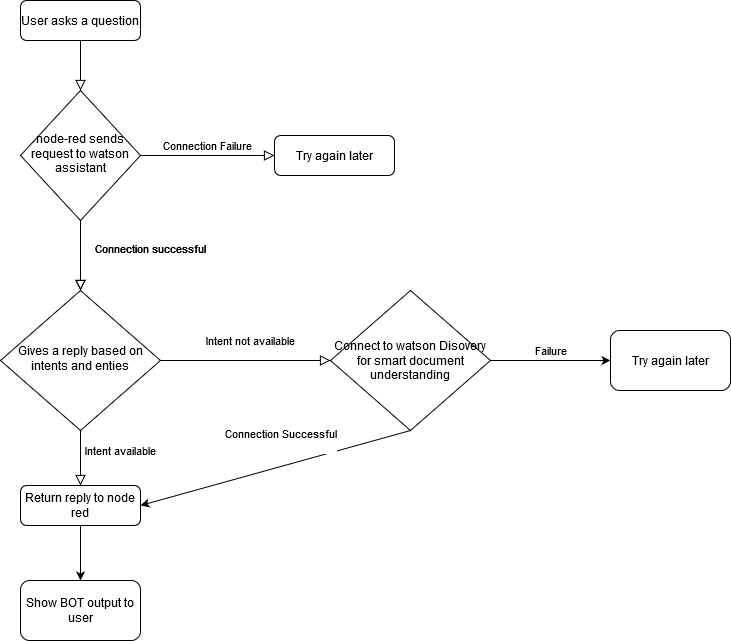
Dialog- Here we mention the outputs to be given, these can be static as well as dynamic.



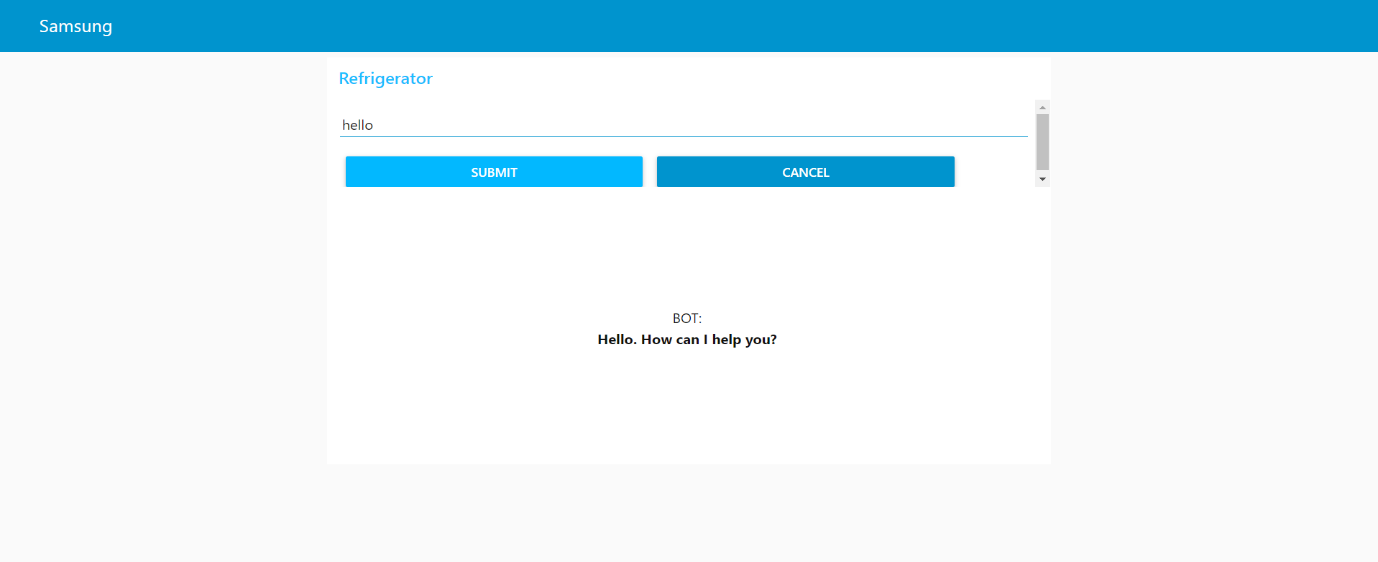
* We have to enable the webhooks which enables our dialog to send a POST request to the webhook URL.
* After this, we have to use Node Red and make a flow to integrate everything. We will use the UI from the node as our front end, we can also use the chatbot from Watson Assistant directly. The roles of different nodes can be understood by the references mentioned in the end. The final flow will look like as shown below.

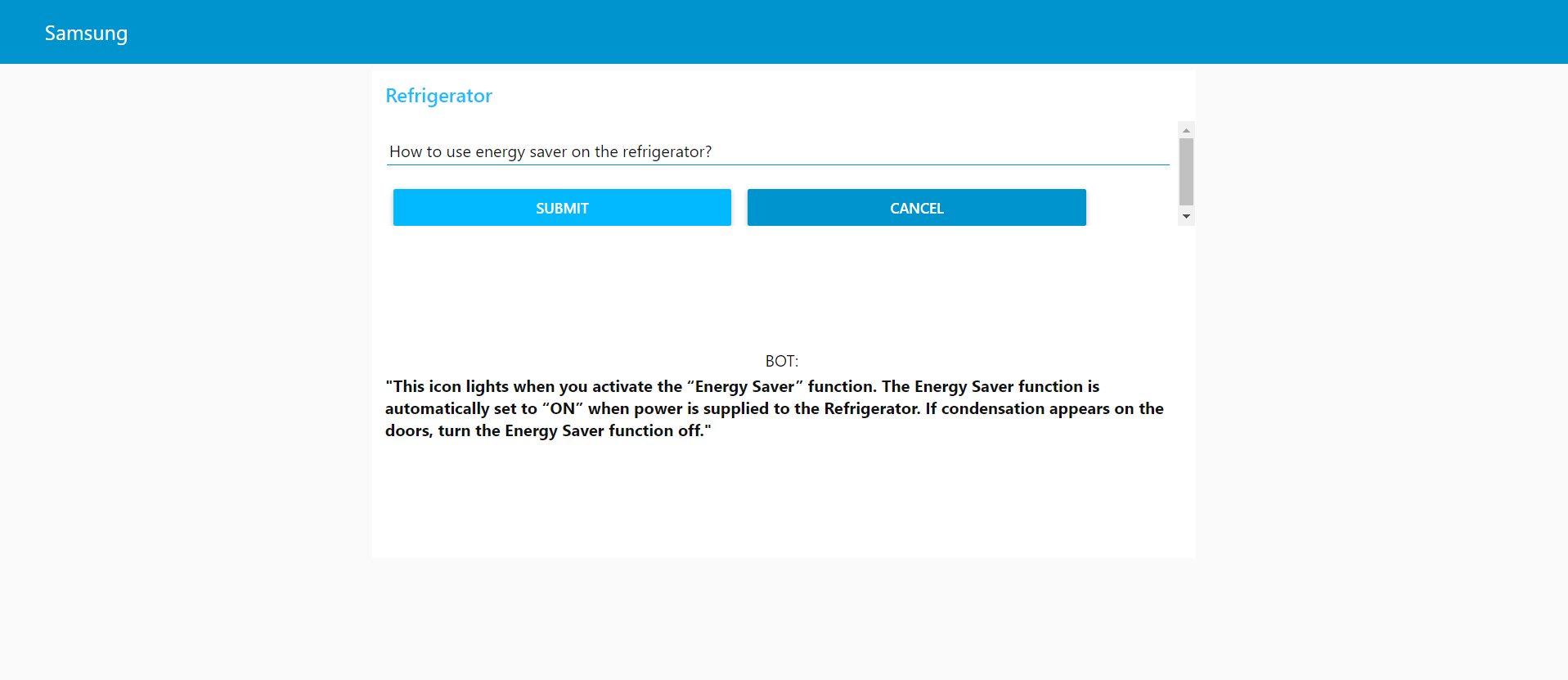


1. **Flowchart:**



1. **Result:**





1. **Advantages and Disadvantages**

**7.1 Advantages**

* Using chatbots saves a lot of time in clarifying user’s queries.
* Far cheaper than hiring customer care executives.
* Customers feel a greater sense of satisfaction as the chatbot helps the user instantly as opposed to being constantly redirected from one executive to another.
* Using chatbots may help you reach more people which will broaden your customer base.
* It doesn’t make the same mistakes humans would have, such as recollection errors, or giving improper information.

**7.2 Disadvantages:**

* Chatbots may provide faster service, but their answers may not be as perfect and specifically tailored for the users’ queries as an executive.
* The chatbots use a limited data base and hence, can’t adapt or improvise. If confused, the conversation may run in loop which can lead to customer frustration and overall dissatisfaction.
* Not all businesses can use them, some businesses are far too complex for chatbots to be practical, as there is no unifying document covering all the important details about the working of the company.
* Performance of the AI relies heavily on the ways it was trained, and if any error, retraining may take significant time.
* The larger the data source, the longer the chatbot would take to train or retrain.
* Unlike humans who could answer questions by searching through a manual simultaneously, a chatbot would need to train on the entire dataset first before answering any user queries.

1. **Applications:**

* Getting quick and precise answers
* Resolving common complaints or problems
* Getting detailed answers or explanations
* Finding a human customer service assistant
* Making a reservation
* Paying a bill
* Finding the location
* Market Research
* Booking flights
* Ordering food

1. **Conclusion:**

By following the basic steps mentioned above, we were successful in creating a chatbot capable on performing basic tasks such as booking appointments, giving store timings and location, as well as answering user queries using Smart Document Understanding from Watson Discovery. We created 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 or rather integrate it with Telegram. We constantly train the bot on current user queries to help improve the speed and accuracy of the results. We can also improve the results of discovery by enriching it with more fields and doing the Smart Data Annotation more accurately, or buy increasing the scope of our chatbot in terms of the calls 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 for Cloud Function**

const assert = require('assert');

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

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);

        });

    });

}

**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>