Intelligent Customer Help Desk With Smart Document Understanding

Project Report :

Project Title : Intelligent Customer Help Desk With Smart Document Understanding.

Domain : Artificial Intelligence.

Contributed by : Deepak Vijay Agrawal.

eMail-ID : debug.career@gmail.com

Table of Contents :

1. INTRODUCTION

1.1. Overview :

We will build a chatbot that uses various Watson AI Services (Watson Discovery, Watson Assistant, Watson Cloud Functions and Node-Red) to deliver an effective 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, IBM Cloud, IBM Watson, Node JS
- Functional Requirements : IBM Cloud
- Technical Requirements : AI, ML, Watson AI, Node JS
- Software Requirements : Watson Assistant, Watson Discovery, Watson Cloud Functions, Node-RED
- Project Deliverables : Intelligent Chatbot with Smart Document Understanding
- Project Deliverables: TheSmartBridge-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.

2. LITERATURE SURVEY

2.1. Existing problem :

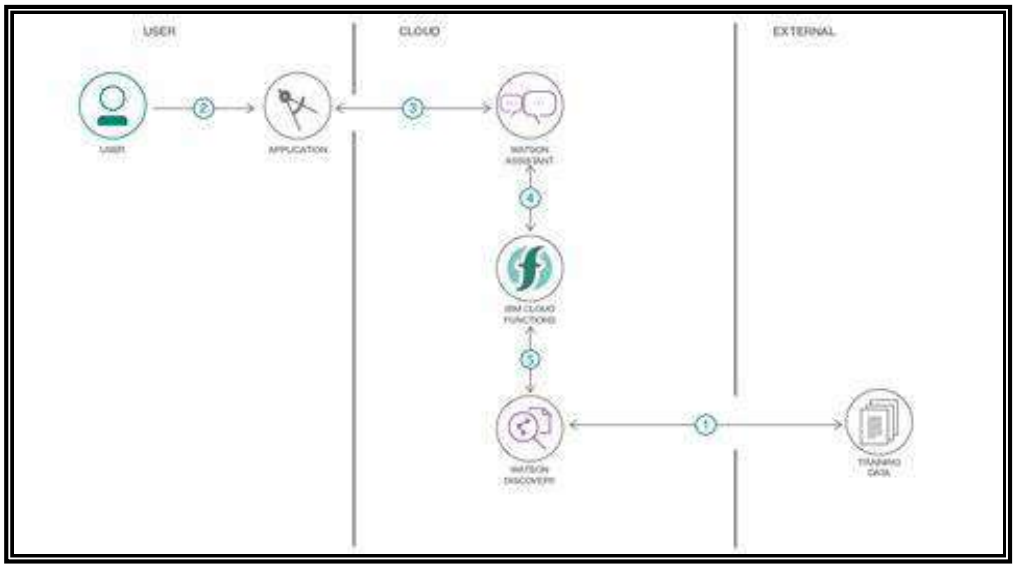
The typical customer care chatbot can answer simple questions, such as store locations and hours, directions, and maybe even making appointments. When a question falls outside of the scope of the pre-determined question set, the option is typically to tell the customer the question isn't valid or offer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

2.2. Proposed Solution :

In this project, 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.

3. THEORETICAL ANALYSIS

3.1. Block Diagram :



2.2. Proposed Solution :

- 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 INVESTIGATIONS

1) Create IBM Cloud Services :

a. Watson Discovery

b. Watson Assistant

c. Node Red

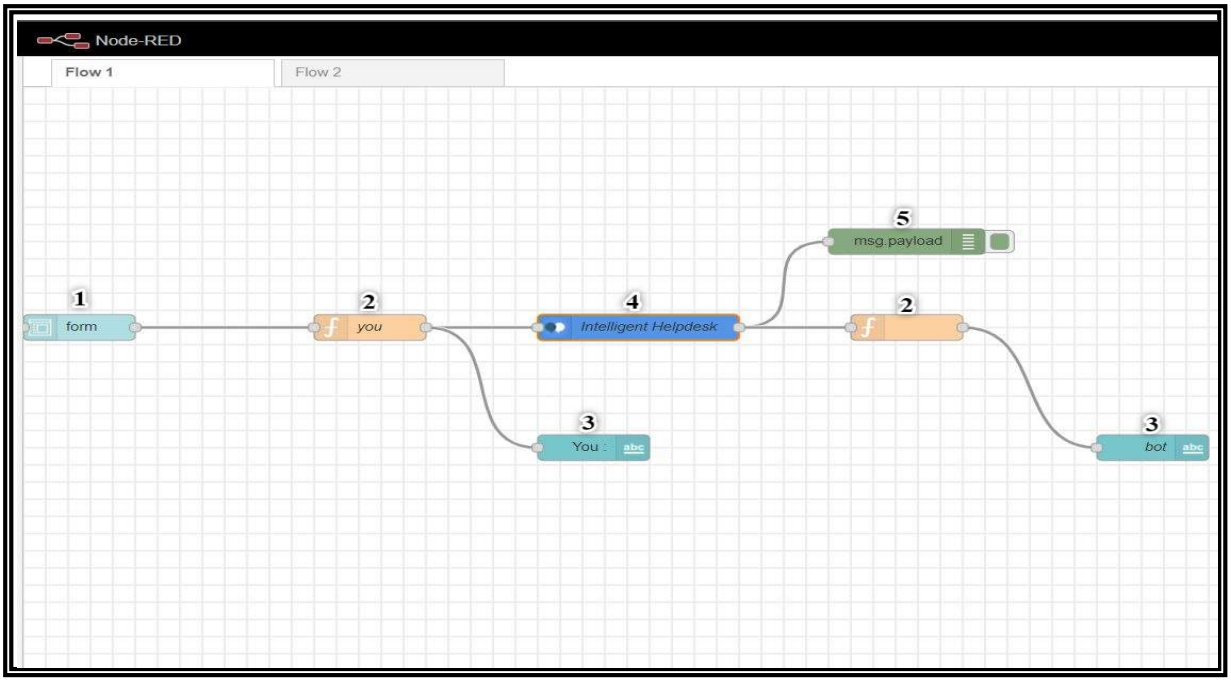
2) Configure Watson Discovery :

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 labeling 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.

5. FLOWCHART :

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

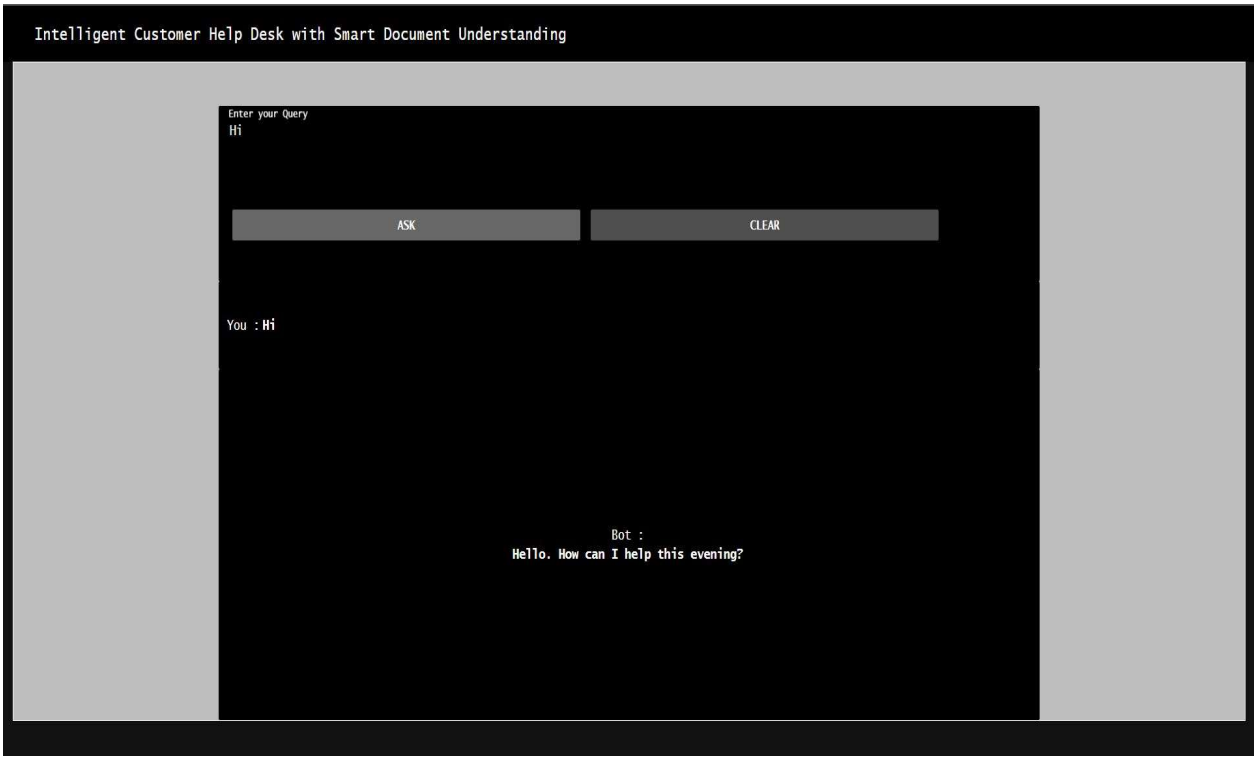
1. **Form.**
2. **Function.**
3. **Text.**
4. **Watson Assistant.**
5. **Debug.**

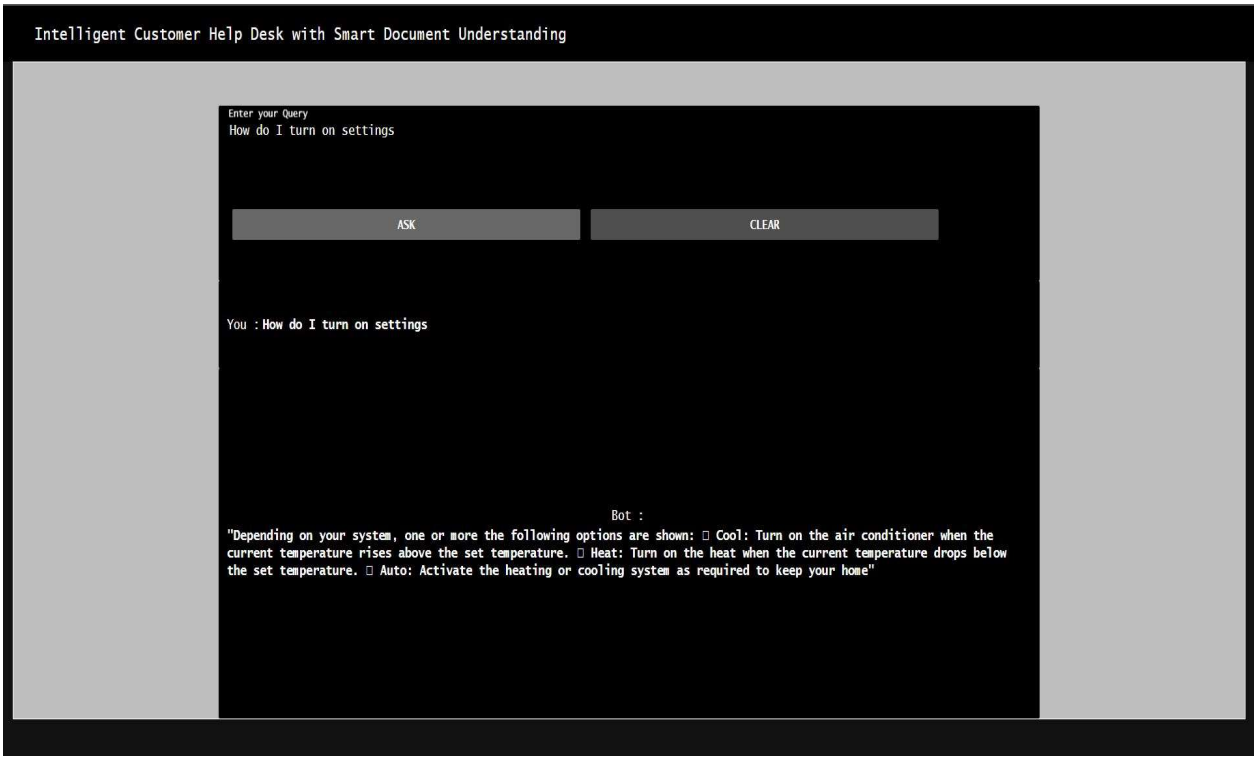


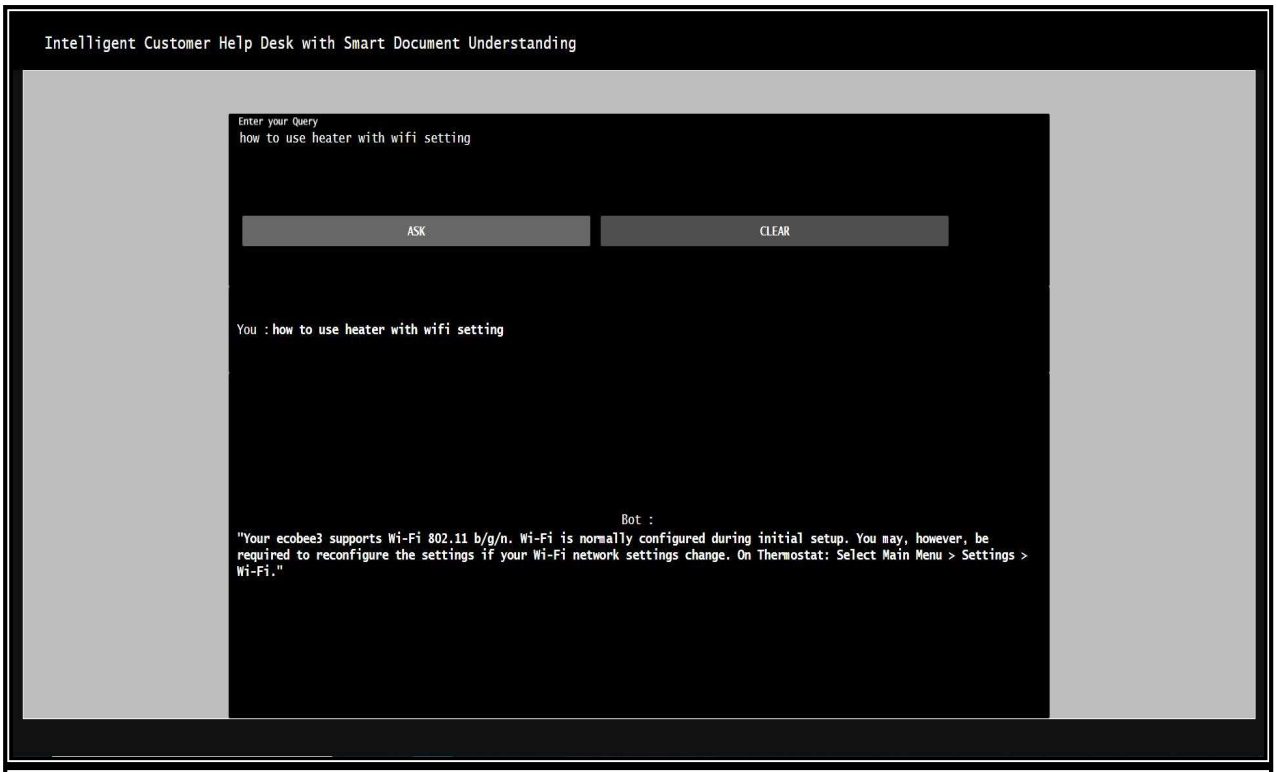
Visit the link below for the Node-Red flow:

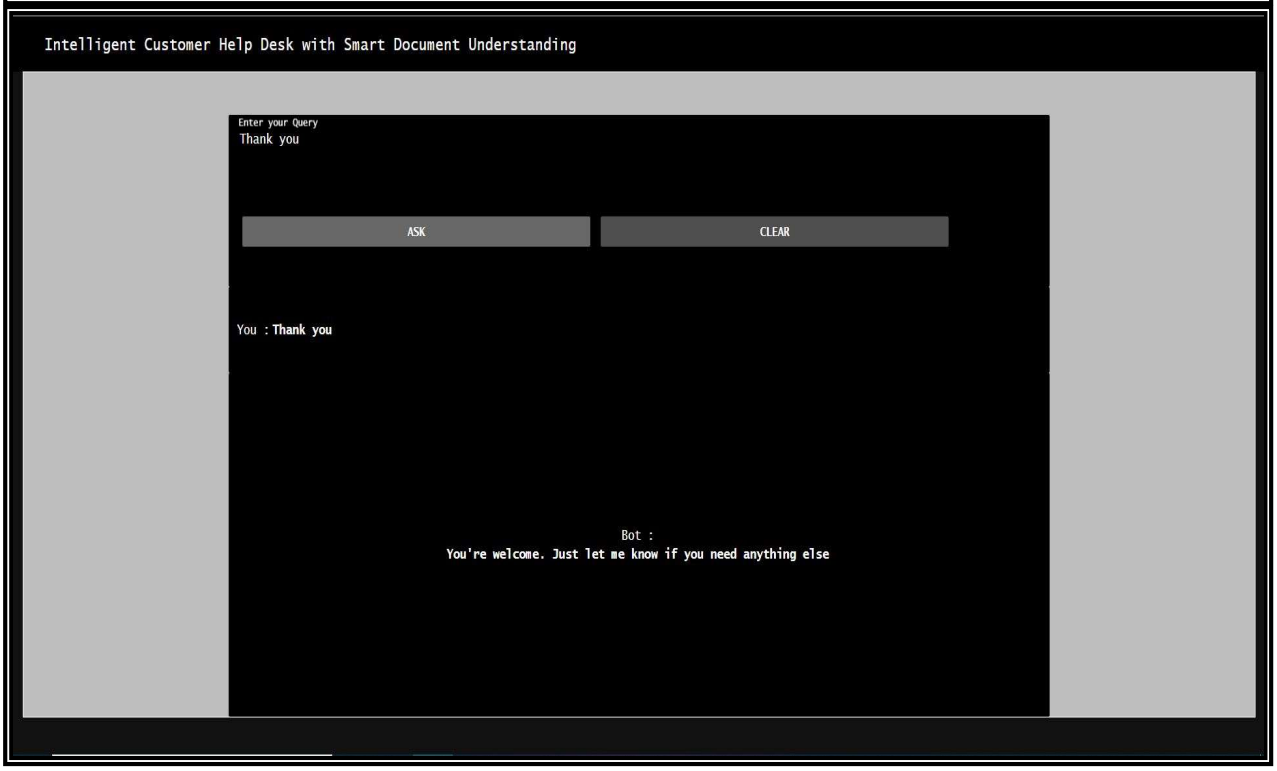
https://helpdesk-nodered.eu-gb.mybluemix.net/red/#flow/26f76e78.f9b3f2

6. RESULT :


Intelligent Customer Help Desk with Smart Document Understanding
Enter your Query
Hi
ASK
CLEAR
You : Hi
Bot :
Hello. How can I help this evening?


Intelligent Customer Help Desk with Smart Document Understanding
Enter your Query
How do I turn on settings
ASK
CLEAR
You : How do I turn on settings
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"





Visit the below link to interact with chatbot :

https://helpdesk-nodered.eu-gb.mybluemix.net/ui/#!/0?socketid=ZE2XAjQ1I406DgeXAAAA

7. ADVANTAGES & 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. APPLICATIONS :

• 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.
• 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 help desk 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 hands free. These are few of the future scopes which are possible.

11.BIBILOGRAPHY

1.Getting started with IBM cloud:

https://www.ibm.com/cloud/get-started

2.Node-Red Starter Application:

https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application

3.Build your own AI Assistant (Chatbot):

https://www.youtube.com/watch?v=hitUOFNne14

4.Watson Discovery :

https://developer.ibm.com/articles/introduction-watson-discovery/

5.Smart Documentation :

https://www.youtube.com/embed/Jpr3wVH3FVA

6.Hands-On with IBM Cloud Function:

https://www.youtube.com/embed/G3bqRndQtQg

APPENDIX :

Source 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.
```

```
25   *
26   */
27 function main(params) {
28   return new Promise(function (resolve, reject) {
29     let discovery;
30     if (params.iam_apikey){
31       discovery = new DiscoveryV1({
32         'iam_apikey': params.iam_apikey,
33         'url': params.url,
34         'version': '2019-03-25'
35       });
36     }
37     else {
38       discovery = new DiscoveryV1({
39         'username': params.username,
40         'password': params.password,
41         'url': params.url,
42         'version': '2019-03-25'
43       });
44     }
45     discovery.query({
46       'environment_id': params.environment_id,
47       'collection_id': params.collection_id,
48       'natural_language_query': params.input,
49       'passages': true,
50       'count': 3,
51       'passages_count': 3
52     }, function(err, data) {
53       if (err) {
54         return reject(err);
55       }
56       return resolve(data);
57     });
58   });
```

```
59 }
```