SMART BRIDGE INTERNSHIP

Intelligent Customer Help Desk with Smart Document Understanding

Project Report

Chatbot for Answering Python Queries

Category: Artificial Intelligence
- By Aditya Mahajan
adityaspmahajan@gmail.com

Index

INTRODUCTION

a) Overview

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 predetermined 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 owner 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.

b) Purpose

While coding there is a lot of time situation like I need to search for various thing for any particular language like we need syntax for a particular part or new thing or we are getting some compile error or there is a keyword which we do not know about and we get stuck and we need to search for that thing on google select the right web page which might contain the solution for which we are looking our answers for.

Also sometimes after a lot of searching, we do not get the solution, so I thought of why not create a bot that would answer the user queries bases on the documentation we provide it. The bot would be trained on the data by splitting the document into various sub-parts based on the structure that we provide in the SDU feature of our training.

LITERATURE SURVEY

a) Existing problem

While coding there is a great deal of time circumstance like I have to scan for different thing for a specific language like we need the functionality for a specific part or new thing or we are getting some incorporate mistake or there is a watchword which we don't think about and we stall out and we have to scan for that thing on google select the correct site page which may contain the answer for which we are looking our responses for.

Every time looking for questions on google is time-consuming and irritating and we can not focus on our main work of developing the feature or doing the project that we want to complete with a specific deadline, which causes a delay in the complete life cycle of project development and our milestones are not achieved and are delayed.

b) Proposed solution

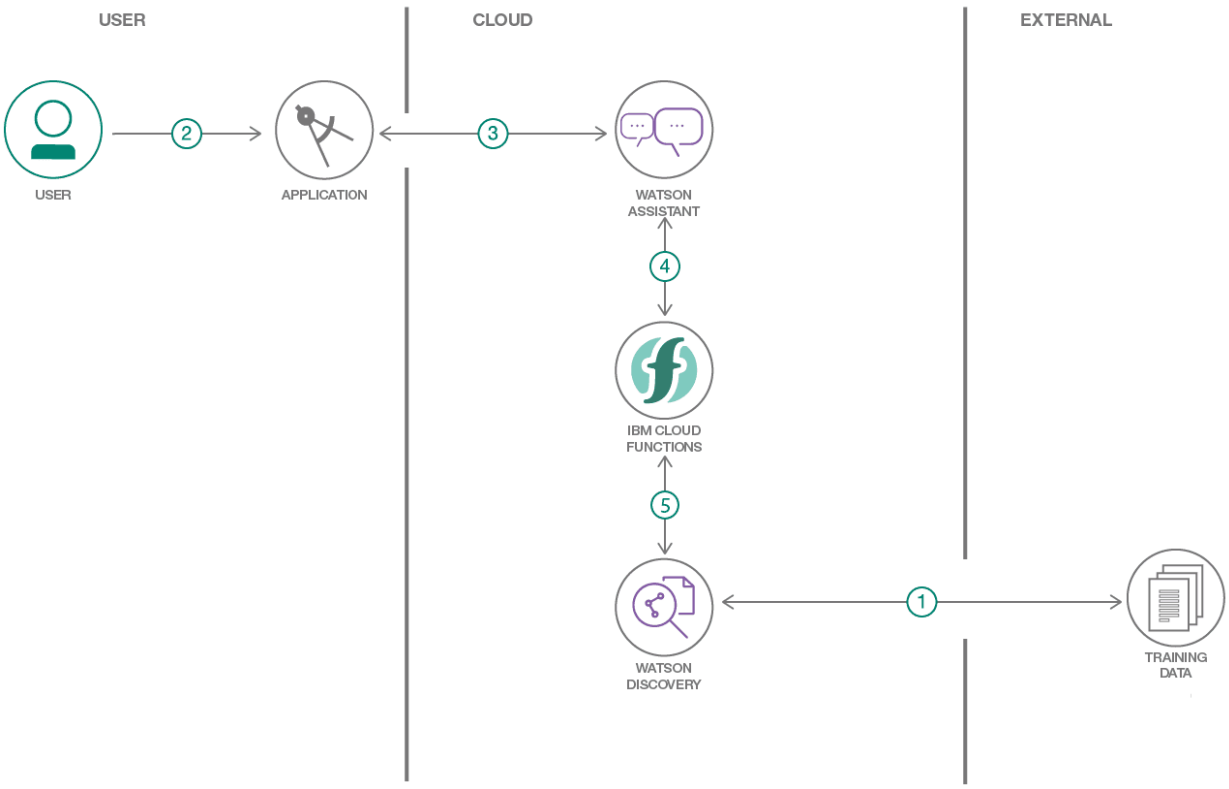
We will be creating a Machine Learning bot that would be able to give a solution to our queries for our python questions. We will be using IBM Cloud services for various things that we need to achieve. The main services that we would be using are as follows:

- Watson Discovery Service
- Cloud Functions
- Watson Assistant
- Node-Red Application

The above mentions services would be used for various things and in the end, we would be creating a simple UI dashboard. In the UI dashboard, we will keep the basic thing that is to ask queries to our bot which would be the endpoint for our bot to give all the results that it would process from the things that it has understood smartly from the training documents it has been given.

THEORETICAL ANALYSIS

a) Block diagram



b) Hardware / Software designing

What is the IBM Cloud platform?

The IBM® cloud platform combines a platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud™ spins up fast and performs reliably in a tested and supported environment you can trust.
https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform

Create a Node-RED Starter Application

Create a Node-RED Starter application running in the IBM Cloud
Secure the application
Customize the Node-RED Starter Kit by adding additional nodes
Go through the Node-RED Starter Tutorial in References.

IBM Watson Platform

Infuse Watson into your apps and workflows to tap into organizational data and put AI to work across multiple departments – from finance to customer care, to supply chain. With Watson, you can create better, more personalized experiences for customers, scale the expertise of your best people across the organization, and make smarter decisions based on deep insights from data.

Introduction to Watson Assistance

Watson Assistant is a conversation AI platform that helps you provide customers fast, straightforward, and accurate answers to their questions, across any application, device or channel. By addressing common customer inquiries, Watson Assistant reduces the cost of customer interactions, helping your agents focus on complex use cases – not repetitive responses.

Introduction to Watson Discovery

With IBM Watson Discovery, you can ingest, normalize, enrich, and search your unstructured data (JSON, HTML, PDF, Word, and more) with speed and accuracy. It packages core Watson APIs such as Natural Language Understanding and Document Conversion along with UI tools that enable you to easily upload, enrich, and index large collections of private or public data.

IBM Cloud Functions

IBM Cloud Functions is a distributed computing service that executes application logic in response to requests from the web or mobile apps. You can set up specific actions to occur based on HTTP-based API requests from web apps or mobile apps, and from event-based requests from services like Cloudant.

EXPERIMENTAL INVESTIGATIONS

1. Create IBM Cloud services

Create the following services: Watson Discovery, Watson Assistant

2. Configure Watson Discovery

Import the document
Annotate with SDU
Store credentials for future use

3. Create IBM Cloud Functions action

Start the IBM Cloud Functions service by selecting Create Resource from the IBM Cloud dashboard. From the Functions' main panel, click on the Actions tab. Then click on Create. From the Create panel, select the Create Action option.

Add the following keys:

- url
- environment_id
- collection_id
- iam_apikey

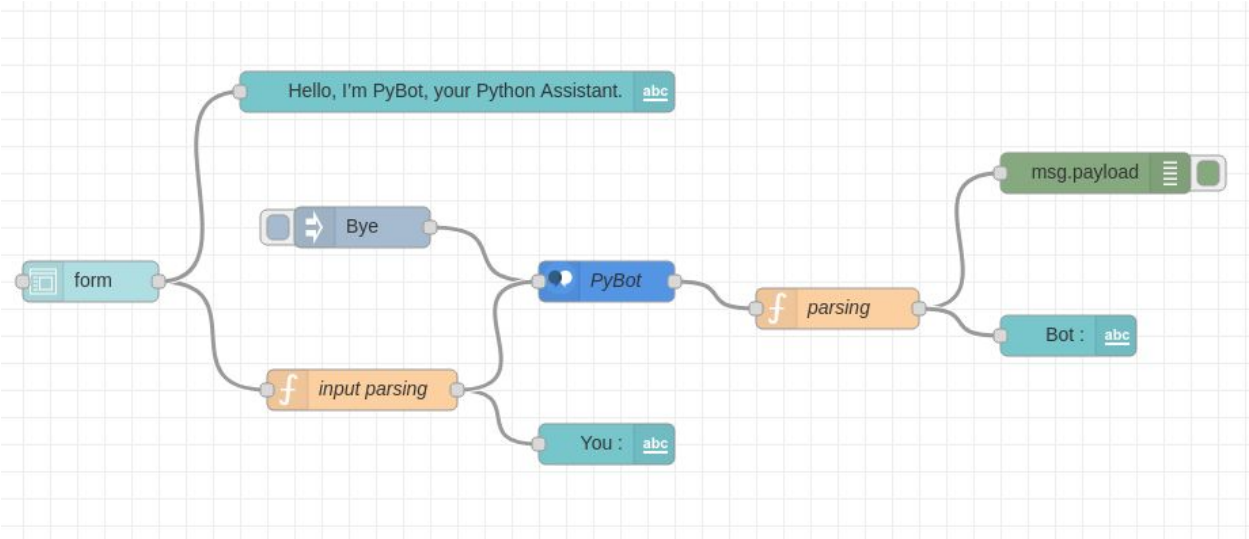
4. Configure Watson Assistant

- **Add new intent**
- **Create new dialog node**
- **Enable webhook from Assistant**

Important: Add .json to the end of the URL to specify the result should be in JSON format.

- **Test in Assistant Tooling**

FLOWCHART



The Flow for the project and the steps that we need to complete the project are as follows:

a) Create necessary IBM Cloud Services
b) Configure Watson Discovery Service
c) Create Cloud Functions Action
d) Configure Watson Assistant
e) Build Node-RED Flow to Integrate All Services
f) Build a Web Dashboard
g) Test the Bot & Capture the Results

Pros

- The main advantage offered by chatbots from the point of view of customer service is automation.
- It can help you establish your online presence outside of your working hours. If a client has a question in the middle of the night, and your business does not provide 24/7 customer support, a chatbot is a solution.
- Humans have a limit to the number of clients they can handle at once. However, with chatbots, there is no such constraint and they can handle as many queries as required at once.

Cons

- The main drawback of the use of chatbots in this context is that the user can easily detect the presence of robotic answers.
- It may cost quite a lot to make a chatbot as specialists in the area are not that easy to find. This especially relates to applications based on AI.
- It is one of the significant limitations of chatbots. These chatbots are programmed in a way that they only know what they are taught. They cannot understand the context of humans, and this is a massive gap that can even lead to an irate customer.

APPLICATIONS OF CHATBOTS

#1. Content delivery

Media Publishers have realized that chatbots are a powerful way to engage with their audiences and monitor engagement to gain valuable insights on reader interests. Chat with the CNN and Wall Street Journal Chatbots on Facebook Messenger and receive the latest news directly in Messenger, without having to visit their websites.

#2. Order Food

Various fast food giants like KFC and Pizza Hut have invested in Chatbots that enable customers to place their orders through conversations. Taco Bell went a step further to improve the conversational experience by giving their Chatbot named TacoBot some personality. It cracks jokes, uses emojis, answers trivia questions, and will even add a cup of water to an order if the customer mentions being hungover.

#3. Book Flights

Icelandair's chatbot gives its customers the ability to search for and book flights in a text-based conversational manner. Instead of drop-down menus, customers enter the information themselves. These features give customers more control over how the flight is booked and it also keeps the entire conversation in one thread so that the purchase information can be reviewed and called up with ease.

#4. Companionship

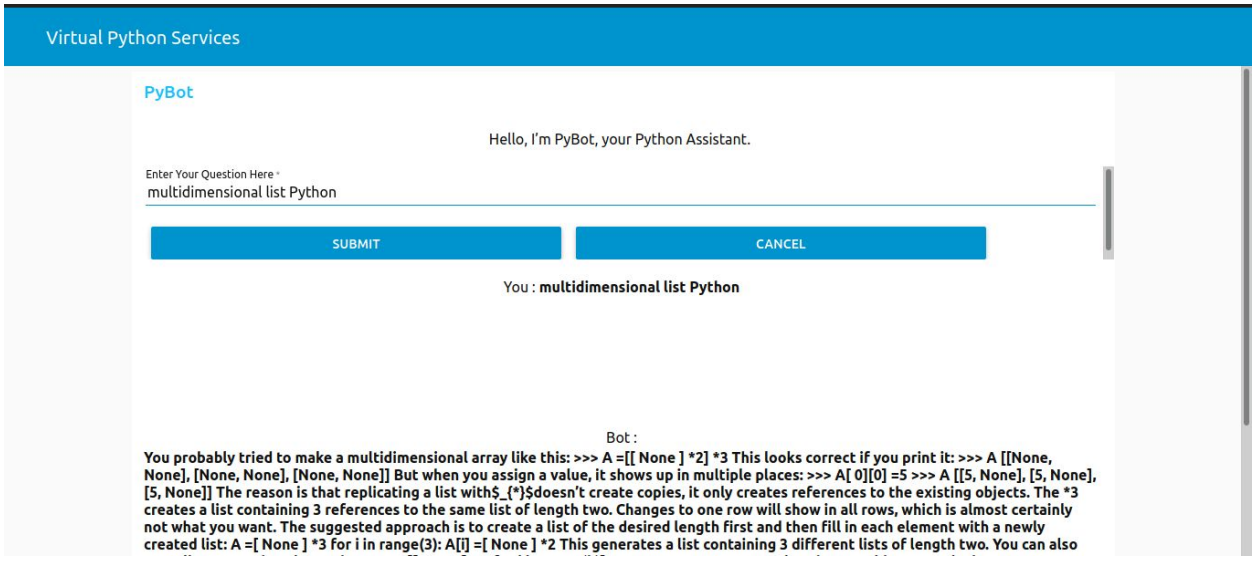
Russian technology company Endurance developed its companion chatbot for Senior People and Patients with Alzheimer's Disease. The primary function of the chatbot is to be a virtual companion – To speak with senior people on general topics like the weather, nature, hobbies, movies, music, news, etc. The chatbot asks questions, reacts to the answers, is able to speak on various topics, and share interesting news and facts from Google.

#5. Transportation

Uber in partnership with Facebook has enabled users to sign up for Uber and request a ride, without having to leave Messenger or download the Uber app. Ride status updates and ride receipts are delivered to a private conversation between the customers and Uber on Messenger, making it easy to track Uber rides and payment history.

RESULT

After Completing the Node-Red Integration and building the dashboard we will have a complete app in which we can ask our python queries.



Node-Red (UI) - https://noderedfirstapplication.mybluemix.net/ui/

Node-Red (Flow) - https://noderedfirstapplication.mybluemix.net/red/

CONCLUSION

In this project, we have established the basis of an automated Chatbot based on AI which examines an uploaded document side by side for efficient and variety of results which increases the precision of the so procured, output provided by the Chatbot. Further, more we have concluded that how an amalgamation of programs has made the result of the project possible, therefore stating that these software's have more horizon to reach than we suggest.

FUTURE SCOPE

I have mainly focused on the functionality part of the project, the main thing that can be improved is UI. Next thing is that this application was made just for experiment and node-red is not very scalable and has very limited functionality so we can shift from node-red to NodeJS or maybe towards python frameworks like Django or flask.

The snippet results are also not very organized showing in the UI which can be improved.

APPENDIX - Source code

Integrate_Discovery.js

```
/**
*
* @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);
  });
 });
}
```

Skill-Customer-Care-Sample-Skill.json

https://github.com/SmartPracticeschool/llSPS-INT-1855-Intelligent-Customer-Help-Desk-with-Smart-Document-Understanding/blob/master/Assesment%20Files/Watson%20Assistant%20Skill/skill-Customer-Care-Sample-Skill.json