Smart Bridge Internship

Project Report

on

AI Powered News Search App

by

SHANJAY KM

KPR Institute of Engineering and Technology,
Coimbatore.

Table of Contents

1. Introduction..3
    a. Overview
    b. Purpose
2. Literature
   Survey...4
    a. Existing Problem
    b. Proposed Solution
3. Theoretical
   Analysis..5
    a. Block Diagram
    b. Hardware/Software Designing
4. Experimental Investigations..24
5. Flowchart...24
6. Result...25
7. Advantages and Disadvantages..26
8. Applications..26
9. Conclusion..26
10. Future Scope..27

11.Bibliography...27

Appendix...28

i. Source code

Github: https://github.com/KMShanjay/SmartIntern

Introduction

1. Overview

The web is home to massive amounts of data, with more being created every day. Organizations can harness this constant stream of information to gain understanding, plan strategies, and find opportunities. Enriched news data can help your application make dynamic connections across current events faster.

2 .Purpose

The purpose of this project is to develop a web application that meets our need to find the most relevant and recent new articles worldwide and update them regularly.

Since the discovery service is integrated with Slack, it gives a bot to search news with a keyword.
In addition, the web application also analyzes the sentiment present in this news article and extracts key words and concepts to make it easier for the user to decipher what is important and what is not.

Literature Survey

a. Existing Problem

All current news applications can be confusing for the average person, with multiple functions and exaggerated design, these applications still do not meet the demand of the latest news users and often get results from the past days, weeks and months, which confuses the user only.

Also, there is no way in these apps to know what the approximate feeling of the audience is regarding the article or news topic, which makes it less interactive and easy to use.

b. Proposed Solution

Discovery service available in the IBM cloud, creating a web app to get the latest news results is fast and hassle-free. When integrated with Red Node Flow, the IBM Discovery Service can create a simple, engaging, organized user interface that provides users with relevant news articles as Discovery Service continuously crawls the web for the latest news to provide.

By adding emotional analysis, we make the user interface more interactive and easier to understand.

Theoretical Analysis

a. Block Diagram

The above block diagram tells user interacts with the user interface created through the red node application that investigates a relevant topic about which he would like to receive news.
The user interface integrated with the discovery service will use the service to present the most recent and relevant news articles, corresponding to what the user searched for.
Since slack is also integrated with discovery, we can use our own bot and ask it to search for news by typing in the keyword and the bot searches for news articles related to it.

b. **Software Designing**

The following Task are required to make the complete application with slack bot integration.

Project Tasks

1. Creating and deploying Watson discovery news app locally.
2. Integrating Slack-bot with Watson Discovery.
3. Creating node-red user Interface.
4. Integrating node-red UI with Watson Discovery.

Setting up the development Environtment

1. Installing Git

Install Git on Windows. The most official build is available for download on the Git website. Just go to https://git-scm.com/download/win and the download will start automatically. Note that this is a project called Git for Windows, which is separate from Git itself.

2. Installing Node.js

Introduction

Node.js is a run-time environment which includes everything you need to execute a program written in JavaScript. It's used for running scripts on the server to render content before it is delivered to a web browser.

2.1.1 Download Node.js Installer

In a web browser, navigate to https://nodejs.org/en/download/

The installer will prompt you for the installation location. Leave the default location, unless you have a specific need to install it somewhere else – then click Next.

Verify Installation

Open a command prompt (or PowerShell), and enter the following:

```
node -v
```

The system should display the Node.js version installed on your system

3. Installing Node-RED

Introduction

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

Installing Node-RED as a global module adds the command node-red to your system path. Execute the following at the command prompt:

```
npm install -g --unsafe-perm node-red
```

Adding A collection of Node-RED nodes for IBM Watson services:

```
npm install node-red-node-watson
```

Adding node-red-dashboard module:
This module provides a set of nodes in Node-RED to create a live data dashboard quickly.

```
npm install node-red-dashboard
```

Task 1 Creating and deploying Watson discovery news app locally.

1.1 Creating Watson Discovery service on IBM cloud.

1. Pre-requisites : IBM cloud account.

1. About Watson Discovery: IBM Watson Discovery, we 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.

Step 1: Login to IBM cloud account.
Step 2: Click 'Create resource' on your IBM Cloud dashboard.
Step 3: Search the catalog for Discovery.
Step 4: Click Discovery to launch the create panel.
Step 5: From the panel, enter a unique name, a region and resource group, and a plan type (select the default lite plan). Click Create to create and enable your service..

The next step is to configure your Watson Discovery service.

1.2 Configuring Watson Discovery service on IBM cloud.

Step 1: Find the Discovery service in your IBM Cloud Dashboard.
Step 2: Click on the service and then click Launch tool

The Watson Discovery News data collection is already associated with your service. You'll use this collection as the data source for your app. To access the collection, you must find the **COLLECTION_ID** and **ENVIRONMENT_ID.** To find these values:

Step 3: Click on the collection from the Manage Data panel. In this case, it is named Watson Discovery News.
Step 4: Click on the drop-down icon located in the top right corner of the panel.

1.3 Saving Watson Discovery credentials.

you'll need to add the Watson Discovery credentials to the .env file Later.

Step 1: Locate the service credentials listed on the home page of your Discovery service.

Step 2: click on the download link and save the file.

1.4 Deploying Watson-discovery-news App locally.

Note: Since deploying App on IBM cloud is not possible with Lite account so we deploy locally on your machines.

Step 1: Use the following command to clone the Watson-discovery-news GitHub repository.

```
git clone https://github.com/ibm/watson-discovery-news
```

Step 2 : Use the following command to create .env file

```
copy env.sample .env
```

Step 3: Use the following command to Enter Watson discovery service credentials in

.env file

step to download credentials here

Copy paste the credentials from downloaded file to .env.

```
1. Notepad .env
```

Step 4: Installing packages and dependencies

A package is a folder containing a program described by a package.json file

This command installs a package, and any packages that it depends on.

```
1. npm install
```

Step 5: Running the deployed app.

```
1. npm start
```

sample output:

Task 2 Integrating Slack bot with Watson discovery news app.

To integrate a new Slack Bot into your existing Slack team, navigate to https://<my.slack.com>/apps/manage/custom-integrations, where <my.slack.com> is the Slack workspace you want to customize.

2.1 Creating slack bot

Step 1: From the Cutsom Integrations page, select the Bots option.

Step 2: To add a new bot, select the Add Configuration button.

Step 3: Enter a username for the bot and click Add bot integration.

Step 4: Once created, save the API Token that is generated.

2.2 Configuring slack bot

Step 1: Edit the .env file and enter the Slack Bot API Token saved in the previous step.

```
# Slack
SLACK_BOT_TOKEN=<slack_bot_token>
#note paste token without the brackets
```

Step 2: Restart application

```
1. npm start
```

Task 3 Creating App user interface using node-red

Installation node-red has been done earlier to refer click-here.

Step 1: Running node-red

Once installed as a global module you can use the node-red command to start Node-RED in your terminal. You can use Ctrl-C or close the terminal window to stop Node-RED.

```
1. node-red
```

Step 2: Node-red flow for Watson discovery news

A **flow** is represented as a tab within the editor workspace and is the main way to organize **nodes**. **Flow** tabs.
Drag and drop the nodes from the nodes from palette then connect using wire

Nodes used:

1. Form node
2. Function node
3. Query node
4. Template node
5. gauge node
6. chart node

Step 3: Configuring the nodes

1. Form node

1. Input function node

Query node configuration is shown in the next Task

1. Output function node

Task 4 Integrating Watson discovery service with UI

1. Configuring discovery node

Experimental Investigation

First we use the discovery service to configure and query adding our collection. A red node application is created in which the discovery is integrated and a simple flow of 5 nodes is created to enter the news topic and the results show related news.

Slack then integrates with Watson's discovery service so that news articles can be searched on more than one platform, and finally sentiment analysis is performed on the data / news articles being searched.

Flowchart

Result

Bot on slack:

The Watson discovery application will be able to fetch relevant news articles within milliseconds, provide sentiment analysis on the fetched data, extract features, concepts, and keywords in the data all in the from of a simple UI.

User Interface can be found at t http://localhost:1880/ui/

Advantages and Disadvantages

1. The web application provides interactive sentiment analysis.
2. It can be accessed through more than one platform that is through slack.
3. It collects and delivers the most recent data.
4. It does not have additional features like storing news history.
5. It does not provide a stand-alone app rather uses a web application.

Applications

1. This web applications can be used by any user in need of accurate and fast results.
2. Can be used by firms and organizations.
3. Can be used in stock market to make predictions.

Conclusion

This project gives some basic working knowledge of the Watson Discovery Service and showed you how to use Discovery along with

JavaScript and Node.js to build your own news mining web application.
It also gives insight into real-world applications of AI and helps us understand Slack better.

Future Scope

1. The web application van be integrated with cloud and made into a mobile app to use it on-the-go.
2. Additional sentiments can be added in the UI.
3. Related and trending news topics can be shown to the user.

Bibliography

1. https://github.com/IBM/watson-discovery-news/

1. https://nodered.org/

1. https://slack.com/intl/en-in/help/articles/360002079527-A-guide-to-Slack%E2%80%99s-Discovery-APIs

2. https://www.w3schools.com/

Appendix

Source code

1. Templete node

CSS part

```
<head>
  <style>
  .cont{
     background-color:#5e64ff;
     outline-style:groove;
     }
  .link{
     color:red;
     background-color:#94ffbf;
  }
  .link:hover{
     font-size:large;
  }
div.inner{
  margin:auto;
  width:80%;
  background-color:white;
  border:1px solid black;
}
```

```
p{
   margin: 5%;
 font-weight: bold;
 color: #000000;
}
   </style>
</head>
```

HTML part

```
<body>

   <div class="cont">
   <h3 class="title">>>{{msg.payload[0].title}}</h3>
   <br/>
   <a target="__blank" class="link"   href="{{msg.payload[0].url}}">{{msg.payload[0].url}}</a>
   <br/>
   <br/>
   <div class="inner"><p><b>{{msg.payload[0].text}}</b></p> <br/></div>
   <br/>
   </div>
   <br>
   <div class="cont">
   <h3 class="title">>>{{msg.payload[1].title}}</h3>
   <br/>
   <a target="__blank" class="link" href="{{msg.payload[1].url}}">{{msg.payload[1].url}}</a>
   <br/>
   <br/>
   <div class="inner"><p><b>{{msg.payload[1].text}}</b></p> <br/></div>
   <br/>
   </div>
   <br>
   <div class="cont">
   <h3 class="title">>>{{msg.payload[2].title}}</h3>
   <br/>
```

```html
    <a target="__blank" class="link" href="{{msg.payload[2].url}}">{{msg.payload[2].url}}</a>
    <br/>
    <br/>
    <div class="inner"><p><b>{{msg.payload[2].text}}</b></p> <br/></div>
    <br/>
    </div>
    <br>
    <div class="cont">
    <h3 class="title">>>{{msg.payload[3].title}}</h3>
    <br/>
    <a target="__blank" class="link" href="{{msg.payload[3].url}}">{{msg.payload[3].url}}</a>
    <br/>
    <br/>
    <div class="inner"><p><b>{{msg.payload[3].text}}</b></p> <br/></div>
    <br/>
    </div>
    <br>
</body>
```