

PROJECT REPORT

ON

"AI POWERED NEWS SEARCH

APP"

By

Phushya Mithra Gauri

Contents:

1. Introduction
 - 1.1 Overview
 - 1.2 Purpose
2. Literature Survey
 - 2.1 Existing problem
 - 2.2 Proposed solution
3. Theoretical Analysis
 - 3.1 Block diagram
 - 3.2 Hardware / Software designing
4. Experimental Investigations
5. Flowchart
6. Result
7. Advantages and Disadvantages
8. Applications
9. Conclusion
10. Future scope
11. Bibliography

Appendix

- A. Source code

1. Introduction :

The title “AI Powered News Search App “ delineates that the entire project is completely dependent on the “Internet“. The **Internet** is the global system of interconnected computer networks that uses the Internet protocol suite (TCP/IP) to communicate between networks and devices. It is a *network of networks* that consists of private, public, academic, business, and government networks of local to global scope, linked by a broad array of electronic, wireless, and optical networking technologies. So as per the definition stated above it is the core of the entire project. If we consider the project as the result then the entire internet which is not directly visible or seen at the time of inspecting the project acts the foundation for the entire and also IBM cloud, IBM Discovery, Node-Red, and other services play a crucial role in obtaining “AI Powered News Search App “.

1.1 Overview :

The main theme of the project is to lessen the burden required for finding / searching for particular news. The output of the project is simple UI which accepts the news query from the user and displays the results regarding news and also the details about the author who wrote the article. And the results are selective only regarding news query and doesn't provide any other information/ads. So that the user can read the exact information without any distraction.

We've also included the sentimental analysis so that the user can know the presence of positivity and negativity in the news in which he was searching.

1.2 Purpose :

In order to get any information regarding a topic, we search the web to get

our answers. The methods used nowadays are not yet up to the mark in providing the news regarding the keyword and this also includes many irrelevant data.

Many organizations all around the world need useful data to plan their strategies and find opportunities. So, enriched news data plays a vital role in the efficient functioning of the organizations to gain a good understanding of the current affairs and build dynamic connections for providing better services. This creates a need for an enriched news mining web application to provide the user with the latest news regarding any topic.

For this purpose, we make use of the pre-built Watson Discovery news collection that discovers trends and patterns and provides The latest information about product and brand perception, events, and more.

2. Literature Survey:

The main idea which supports creating this project is to minimize the problems that are prevalent in many news mining applications like retrieving irrelevant data and providing information that is not up to date. The web crawlers download pages that do not update their database from time to time, as a result, it provides outdated information to the users. The Watson Discovery Service is a solution to all these problems as it is updated continuously with new articles from various sources. It also uses techniques like news alerting and event detection to provide the trending news in any field that is being searched. Thus, we can track recent historical trends and stay abreast of the latest information about various products, events, and many more.

2.1 Existing problem :

Here are some of the problems which obstruct the news search engines for

obtaining the 100% accurate results :

- **Relevance :**

You can do a search for 'coffee' in Canada and find Tim Horton's website as the most relevant. It makes sense, as that's the most popular coffee chain in Canada, but for somebody in Seattle, Starbucks might be the most relevant result. You can do a search for the '49ers' and be looking for the football team, but a historian may be looking for research material on California. And you might even do a search today for 'bones' trying to find where to buy your dog a treat, but tomorrow you do that same search looking for an episode of the TV series 'Bones' that you missed the night before.

- **Natural language searches:**

A MySQL database engine can precisely return all the relevant records given a query 'select first, last from employee where last = "Smith";'. There are formal syntax and no ambiguity. A search engine, on the other hand, receives 'who has smith as last name in Chicago' or 'smith last name Chicago'. The query is in natural language/our language. There are many different ways to say the same thing/there is context, and so on. The searcher component of a search engine must disambiguate the query and translate it into a more formal manner before looking it up in the index.

- **Poor queries :**

Many searchers don't know how to express what they want in the real

world and are even worse when attempting to ask a search engine. They call the vacuum cleaner a ‘sucker’ and are unable to find cleaning services online. Worse yet they misspell words, making the problem more ‘interesting’ for search engines.

- **Synonymy :**

This is another challenge. There are words that have the same meaning, like ‘car’ and ‘automobile’. When you do a search you would like to get pages that contain your exact words, and pages that contain other words that mean the same thing, as long as they are relevant to your search. Let’s say you do a search for ‘monkey’. You would want your results to include pages that contain monkey, but perhaps also the words ‘chimpanzee’ or ‘ape’. If you were a little bit more strict, you would not want to include pages that say chimpanzee because, although a chimpanzee is a primate, it is not a monkey. These details don’t pass through the minds of most searchers, but search engines have a hard time because of it.

2.2 Proposed solution :

In this project, the main “ **Aim** “is to provide efficient enriched news data by using “**Artificial Intelligence** “. when we dive deeper the entire project can be broken into separate stages they are :

1. **UI:**

This stage is the user interface that serves as the mediator between the naive user who is trying to find out the news for a respective query and a powerful AI tool for collecting data and refining it. UI is created by the Node-red tool. It also

converts and displays the data collected from IBM Discovery into a human-understandable form.

2. **Data Cleaning and refinement :**

In this stage, the tool works on the query keyword given by the user and gathers all the news regarding that keyword. Here itself it cleanses all the data gathered from different sources and frees the data from spamming and etc.

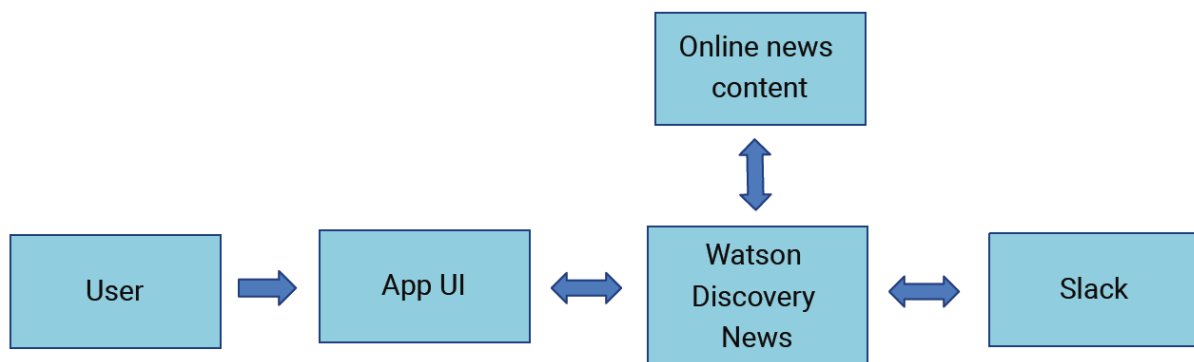
3. **Slackbot ;**

It is an additional resource provided to users for interactive interaction between the user and a system. The slackbot also collects the search query from the user and displays the results.

In the end, by a combination of all these three processes, we try to provide the news to the user with an interactive environment.

3. **Theoretical Analysis :**

3.1 **Block diagram :**



3.2 **Hardware / Software designing :**

There are few hardware and software requirements that are necessary for the execution of the project.

Hardware Requirements:

1. Processor: Minimum 1GHz.Recommended 2GHz or more.
2. Ethernet Connection (LAN) or Wireless adapter (WiFi)
3. Memory (RAM): Minimum 3GB.Recommended 4GB or more.

Software Requirements:

1. Recommended OS: Windows, MAC or Linux
2. Git control system
3. Node.js runtime environment
4. Tools like Slack and Node-Red

4. Experimental Investigations :

For creating a News-Mining App there are three essential tools :

1. **IBM Discovery:** IBM Watson Discovery brings together a functionally rich set of integrated, automated Watson APIs to:
 - Crawl, convert, enrich, and normalize data.
 - Securely explore your proprietary content as well as free and licensed public content.
 - Apply additional enrichments such as concepts, relations, and sentiment through Natural Language Understanding (NLU).
 - Simplify development while still providing direct access to APIs.
2. **Node-Red:** Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. Elements of applications can be

saved or shared for re-use. The runtime is built on Node.js. The flows created in Node-RED are stored using JSON.

3. **Slack:** Slack is a proprietary business communication platform developed by Slack Technologies.

When we get into creation initially we created an account in the IBM cloud and created a Watson Discovery service. After creation, we invoke the Discovery by launching it so that it keeps searching when a query is given to it. By default, the discovery service consists of a predefined dataset that gets updated on a regular basis and we used it. Otherwise, you can define a dataset and upload it to the discovery service. After launching the mining tool, we have used Node-red to create flow-based UI for the user basically, Node-red is a flow-based editor for creating UI's and later we integrated the IBM discovery and UI prepared in Node-red by providing the API key and URL for nodes in Node-red which are provided to us at the time of launching the discovery service and also Nde red contains a wide range of nodes for integrating both node-red and IBM Discovery. Lastly, we created a Slackbot in Slack to make the interaction between the user and System more interactive and interesting. We integrated the Slackbot and Discovery.

5. Flowchart :

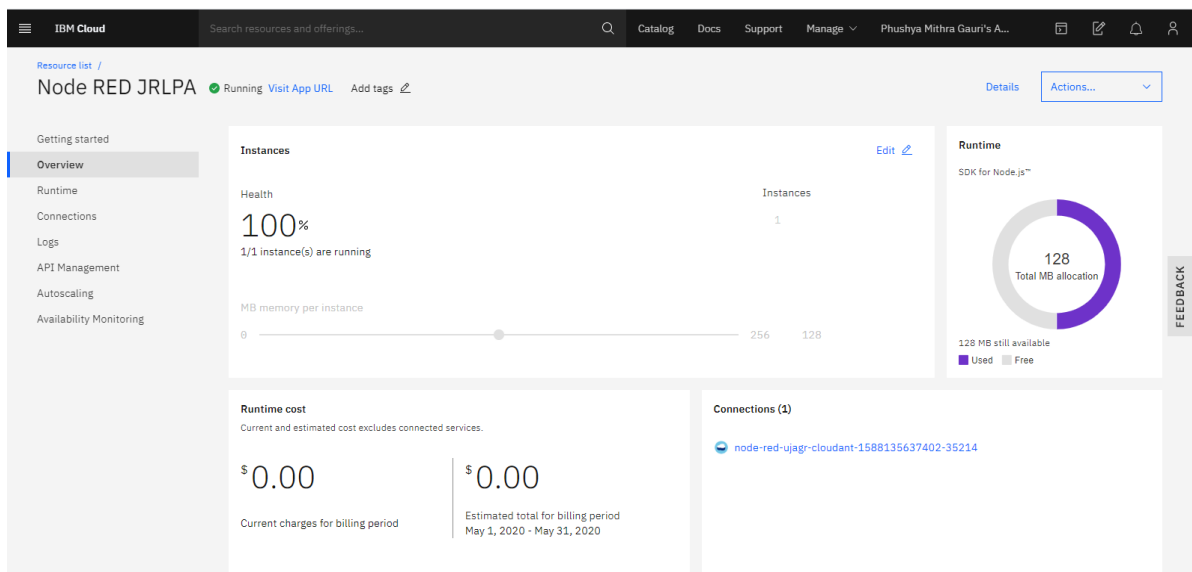
Step-1: Initially, create an IBM Cloud Account by filling out the details and complete the registration by providing your personal details and signing in.



Link: <https://cloud.ibm.com/registration>

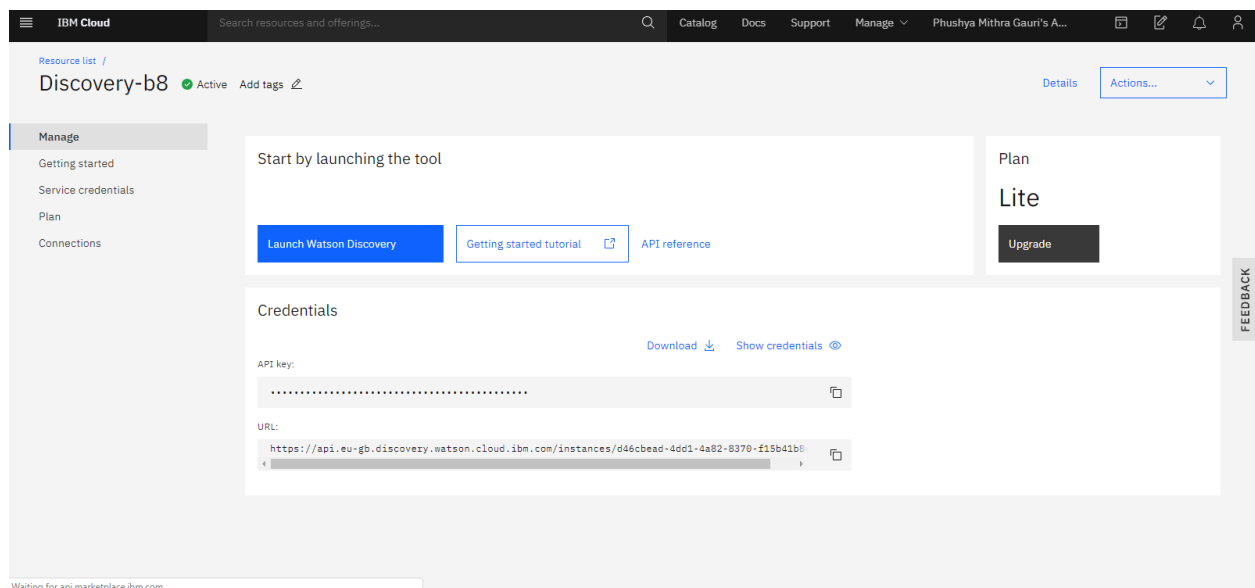
Step-2: Later, create an IBM cloud app by giving the details of the app and also the server details like location, etc. to get started with your custom application. After the app is created, you can find it under the resource list and access it from there.

The app will look like below

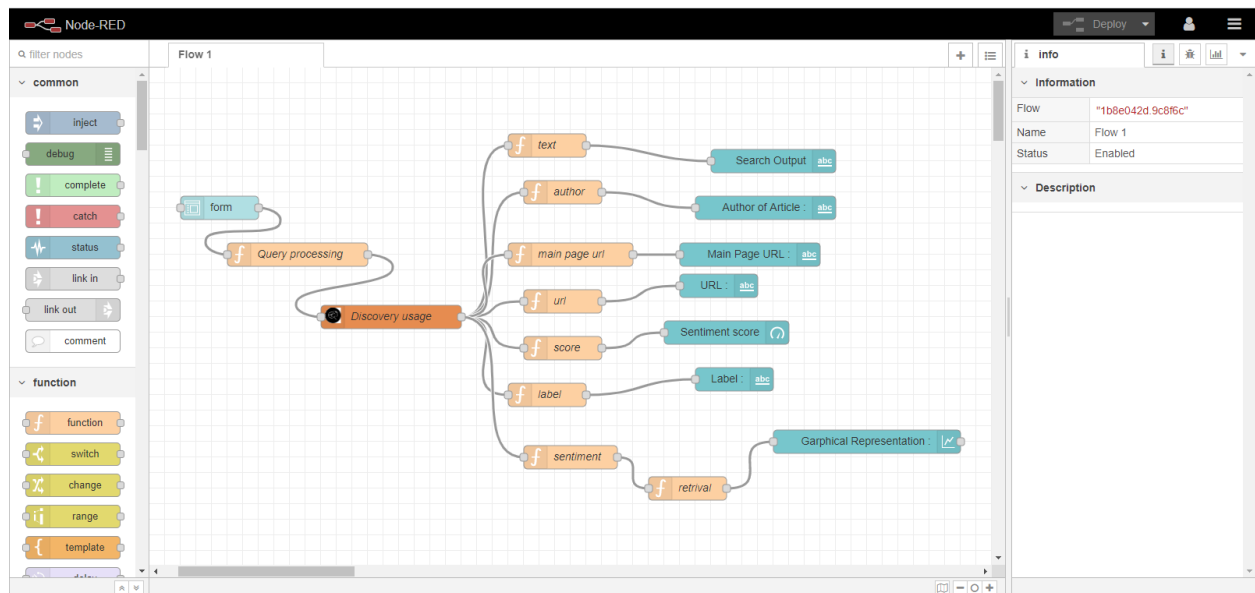


Step-3: Once, the app is created, go to the AI sub-section of the services section under the catalog and select the "Discovery" service. Then, choose the pre-built

Watson Discovery News Collection and then launch the service.



Step-4: After launching the Discovery service, click the URL of the app to create the flow using the Node-Red tool. Go to the Node-Red flow editor and use the nodes to create the flow.



Step-5: Finally, we have to integrate the application with slack by following the below steps.

1. Download Git for windows by using the link below.

<https://git-scm.com/download/win>

And clone the watson-discovery-news Github repository by executing the command git clone <https://github.com/ibm/watson-discovery-news>

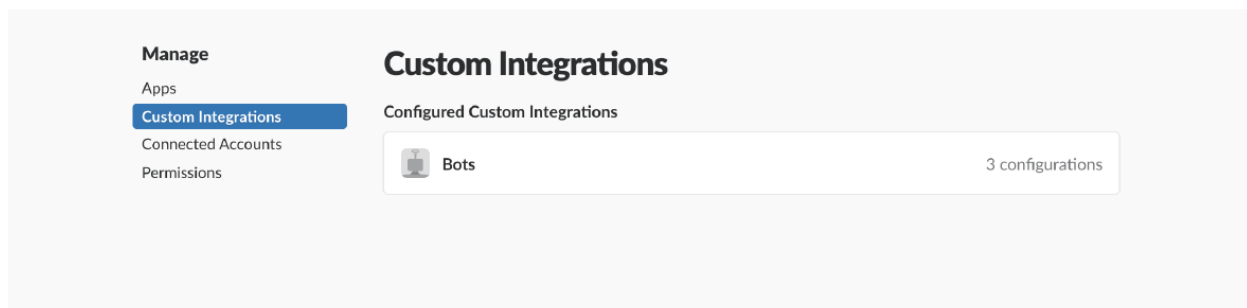
2. From the home directory of your cloned local repo, create a .env file by copying it from the sample version.

copy env.sample .env #Command for Windows user

3. Copy all the service credentials listed on the homepage of your Discovery service and paste it in the .env file.


4. 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.

- From the Custom Integrations page, select the Bots option.



- To add a new bot, select the Add Configuration button.

[< Browse Apps](#)



Add to Slack

App help

Terms

Report this app to Slack for inappropriate content or behavior.

Bots



App Info **Settings** Security & Compliance **NEW**

This app was made by Slack.



This app was made by a member of the Slack team to help connect Slack with a third-party service; these apps may not be tested, documented, or supported by Slack in the way we support our core offerings, like Slack Enterprise Grid and Slack for Teams. You may provide feedback about these apps at feedback@slack.com.

It only uses data Slack already has access to (view our [Privacy Policy](#) to learn more). By enabling and/or using this app, you may be connecting with a service that is not part of Slack.


Configurations

 Connect to Slack as mithra-newsbot
gauriphushyamithra on May 13, 2020 

- Enter a username for the bot and click Add bot integration.

 slack | App Directory [Browse](#) [Manage](#) [Build](#) 

[Browse apps](#) > [Custom Integrations](#) > [Bots](#) > New configuration

 **Bots**
Connect a bot to the Slack Real Time Messaging API.
Run code that listens and posts to your Slack team just as a user would.

Username
Start by choosing a username for your bot

Names must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores. Most people choose to use their first name, last name, nickname, or some combination of those with initials.

Add bot integration

By creating a bot integration, you agree to the [Slack API Terms of Service](#).

- Once created, save the API Token that is generated.

Browse Apps > Custom Integrations > Bots > Edit configuration



Bots

Added by gauriphushyamithra on May 13th, 2020

Disable • Remove

Run code that listens and posts to your Slack team just as a user would.

Setup Instructions

Please refer to our [bot user API](#) documentation, which tells you everything you need to know about setting up a bot integration.

Integration Settings

API Token

The library you are using will want an API token for your bot.

xo xb-1097065191958-1120425924837-cifDNxBWkYUE8TiV2hopssxo

[Regenerate](#)



Be careful when sharing bot user tokens with applications. Do not publish bot user tokens in public code repositories. [Review token safety tips.](#)

Customize Name

Choose the username for this bot.

mithra-newsbot

Username must be all lowercase. They cannot be longer than 21 characters and can only contain letters, numbers, periods, hyphens, and underscores. Most people choose to use their first name, last name, nickname, or some combination of those with initials.

5. Edit the .env file and enter the Slack Bot API Token saved in the previous step.
6. Finally, run the application by executing the following commands.

```
npm install
```

```
npm start
```

```
npm
Microsoft Windows [Version 10.0.18362.778]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Captain CW10>cd watson-discovery-news

C:\Users\Captain CW10\watson-discovery-news>npm start

> watson-discovery-news@1.0.0 start C:\Users\Captain CW10\watson-discovery-news
> node --max_old_space_size=512 app.js

Initializing Botkit v0.7.6
info: ** No persistent storage method specified! Data may be lost when process shuts down.

*****
* WARNING: Your bot is operating without recommended security mechanisms in place. *
* Initialize your Botkit controller with a clientSigningSecret paramter to enable *
* verification that all incoming webhooks originate with Slack: *
* *
* var controller = new Botkit.slackbot({clientSigningSecret: <my secret from slack>}); *
* *
*****
>> Botkit docs: https://botkit.ai/docs/readme-slack.html#securing-outgoing-webhooks-and-slash-commands
>> Slack docs: https://api.slack.com/docs/verifying-requests-from-slack

Watson Discovery News Server running on port: 3000
notice: ** BOT ID: mithra-newsbot ...attempting to connect to RTM!
notice: RTM websocket opened
```

6. Result :

Images of the UI output:

News search APP

Using WATSON

Query for news *

cars

SUBMIT

CANCEL

Author of Article : Nick Broutin

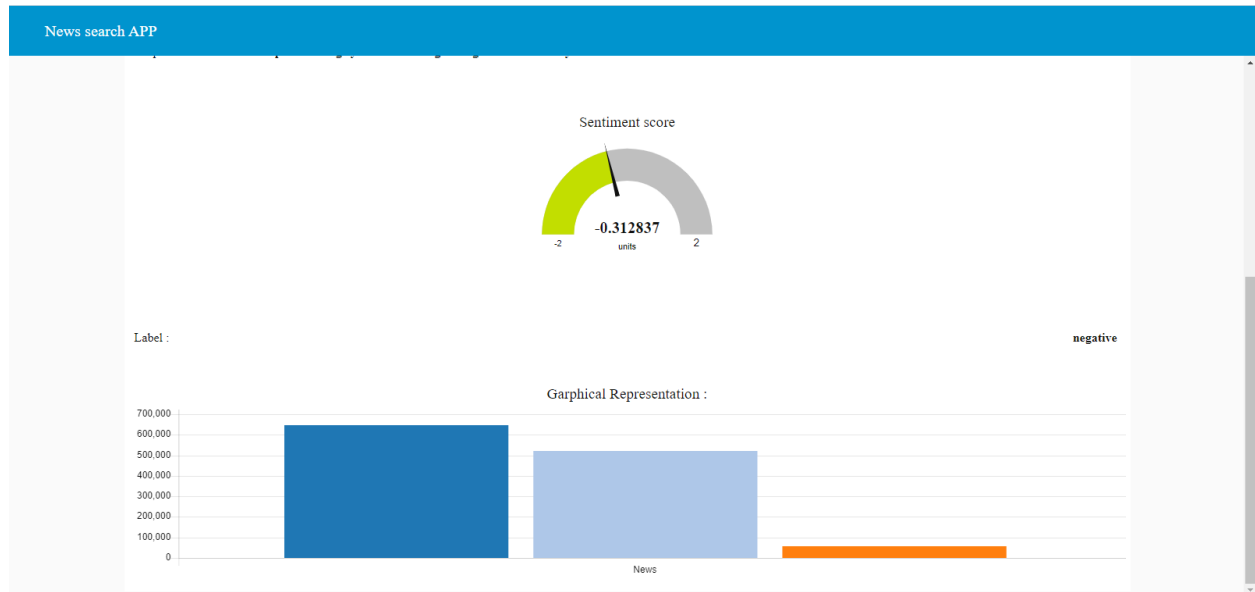
Main Page URL : https://www.autoloanseastcoast.com

URL : https://www.autoloanseastcoast.com/used-cars-in-brooklyn-new-york/easily-find-a-used-car-at-a-used-car-dealership-in-brooklyn-new-york/

Search Output Easily Find A Used Car at a Used Car Dealership in Brooklyn New York on November 16, 2019 One of the easiest ways to find a pre-owned car is to shop online but don't let the car dealerships discourage you from finding the right used car that you need in

Sentiment score

0.317837



URL Address of the UI:

<https://node-red-jrlpa.eu-gb.mybluemix.net/ui>

Images of the Slack bot output:



mithra-newsbot ☆



Messages 1 About



gauriphushyamithra 11:07 PM

hi



↓ 1 new message ×



mithra-newsbot APP 11:07 PM

Hello.



gauriphushyamithra 11:07 PM

can you find news for me ?

news please



mithra-newsbot APP 11:08 PM

Hi there!

What news are you interested in?



gauriphushyamithra 11:08 PM

cars



mithra-newsbot APP 11:08 PM

You want me to search for news articles about cars ?



gauriphushyamithra 11:08 PM

yes

Message mithra-newsbot



B

I

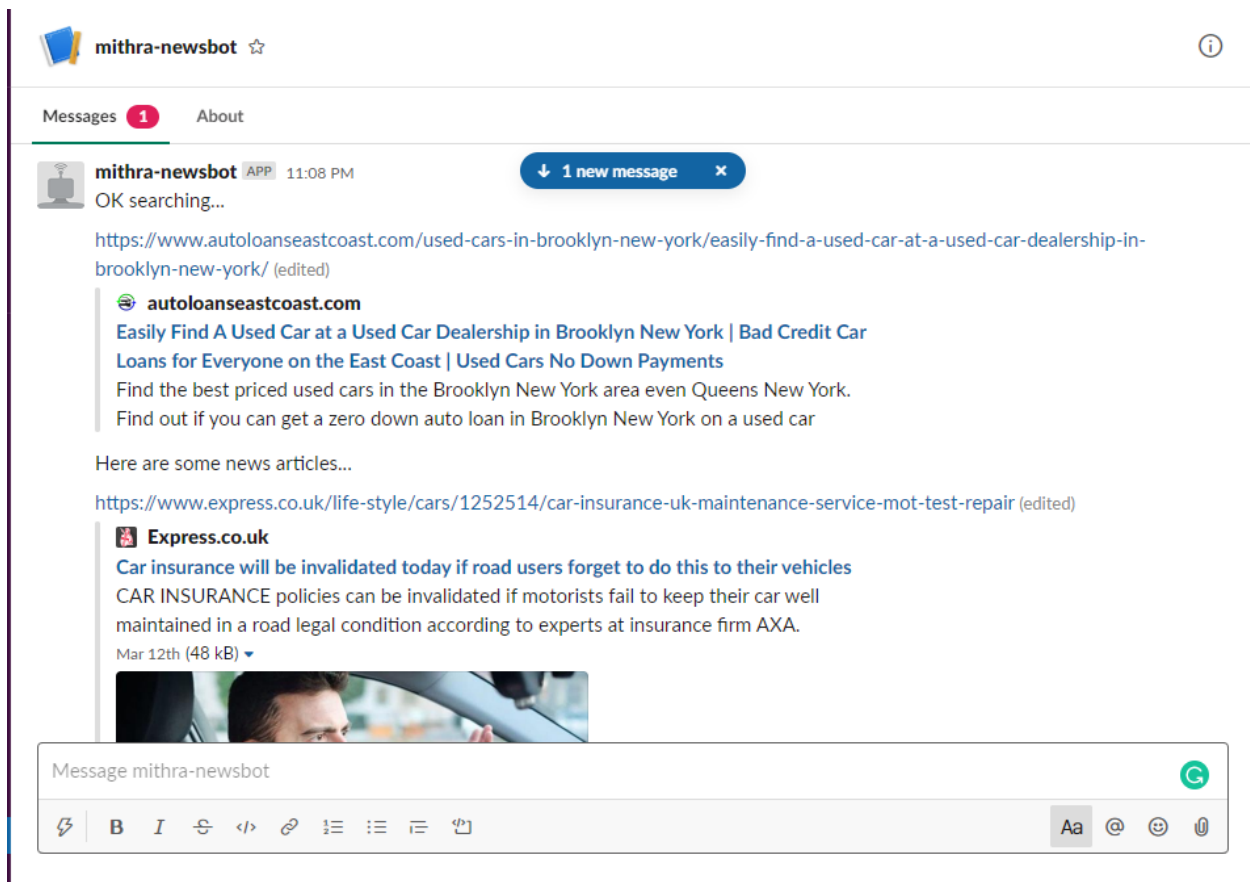


</>



Aa





7. Advantages and Disadvantages :

Advantages :

- As this news app created in this project is powered by node-red it would be easy to change the theme or font or process as it's composed of modules of code that are already present. The flows created in **Node-RED** are stored using JSON which can be easily imported and exported for sharing with others.
- The root of gathering news is IBM's prestigious "Watson Discovery", which is robust.
- Watson Discovery's Integrated Machine Learning capabilities allow it to continuously improve from user interactions and enable users to create and

deploy custom machine learning models based on domain expertise faster than ever before.

- Watson Discovery's integration with IBM's award-winning conversational AI.

Disadvantages

- Node server comparing to Java server has a slower performance.
- And for Watson, there are some disadvantages like :
 - Only in English (Limits areas of use)
 - Seen as a disruptive technology
 - Maintenance
 - Doesn't process structured data directly
 - Takes time to integrate Watson and its services into a company
- Up to now, the designed page is not yet responsive.
- So, the user may encounter many implications in reading the content present on the webpage through different devices.

8. Applications :

The main application of this project is to provide enriched news data about any particular topic. It not only helps to gain an understanding of it but also proves helpful to make dynamic connections across the events in a faster way. The user can query for the most relevant news articles about a specific topic or subject. Because the news collection is pre-enriched with natural language processing, you can query not just on keywords or categories but also on concepts, sentiment, and relations to get richer search responses. We can also identify popular topics over the past 24 hours. Topics can be general or specific to an industry or category. The Watson Discovery service also identifies important meta-information – like authors, publication dates, and relevant keywords in addition to sentimental

analysis of the data. By discovering trends and patterns in sentiment with aggregate analysis, you'll see new perspectives on how news unfolds across the globe. We can even create news alerts by taking advantage of the support for entities, keywords, categories.

9. Conclusion :

This project gives you some basic working knowledge of the Watson Discovery Service and shows you how to use Discovery along with JavaScript and Node.js to build your own news mining web application. With open-source toolkit Botkit, it's easier than ever to create your own chatbot to respond to the user by giving them all the latest news. In this project, which demonstrates how to serve news articles with the Watson Discovery API, we use Botkit to create a chatbot that acts as a middleman to facilitate queries to the Watson Discovery Service. This project satisfies its objective of creating a news mining application that provides the user with all the latest trends and patterns.

10. Future scope :

This application can further be improved by including multiple language querying. That is, the user can ask queries in multiple languages and the application can retrieve the data by interpreting the query. We can also include a customer feedback option to make changes to the UI and make it much better. The application can even be integrated with Watson's Assistant so that chatbots can access the Discovery service and reply to the user's questions. Even, the layout of the UI can be optimized to make it more simple and clean. Thus, making the UI more attractive. We can also develop a more Mobile-optimized design so that the users can access the application even on their mobiles. The load time of the graphics can also be decreased by minifying the page and making the user

experience far more pleasurable.

11. Bibliography :

References:

1.

<https://developer.ibm.com/technologies/artificial-intelligence/patterns/create-acognitive-news-search-app/>

2. <https://www.ibm.com/watson/services/discovery-news/>

3.

<https://developer.ibm.com/tutorials/integrating-slack-with-watson-discovery-news/>

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

Appendix

A. Source code :

```
[{"id":"1b8e042d.9c8f6c","type":"tab","label":"Flow
1","disabled":false,"info":""},{ "id":"63491e76.16b92","type":"ui_form",
"z":"1b8e042d.9c8f6c","name":"","label":"","group":"2f3a0665.46bd5a"
,"order":1,"width":"0","height":"0","options":[{"label":"Query for
news","value":"Input","type":"text","required":true,"rows":null}], "form
Value":{"Input":""},"payload":"","submit":"submit","cancel":"cancel","t
opic":"","x":190,"y":180,"wires":[["3b99bd99.e08182"]]}, {"id":"3b99bd
99.e08182","type":"function","z":"1b8e042d.9c8f6c","name":"Query
processing","func":"msg={\n\ndiscoveryparams:\n
{\n\"environment_id\": \"system\", \n\"query\": msg.payload.Input\n
}\n}\nreturn
```

```
msg;", "outputs":1, "noerr":0, "x":290, "y":240, "wires":[[{"id":"8adf8da.191487"}]], {"id":"5fa4960f.888e68", "type":"function", "z":"1b8e042d.9c8f6c", "name":"text", "func":"msg.payload=msg.search_results.results[0].text\nreturn
```

```
msg;", "outputs":1, "noerr":0, "x":610, "y":100, "wires":[[{"id":"3bf056da.b636ba"}]], {"id":"8adf8da.191487", "type":"watson-discovery-v1", "z":"1b8e042d.9c8f6c", "name":"Discovery
```

```
usage", "environmentname":""," "environmentId":"system", "collectionId":"news-en", "configurationname":""," "configurationId":""," "language_code":"en", "collection_name":""," "count":"1", "passages":true, "nlp_query":true, "query":""," "filter":""," "aggregation":"term(enriched_text.document.label,count:10)", "return":""," "description":""," "size":"LT", "discovery-method":"query", "service-endpoint":"https://api.eu-gb.discovery.watson.cloud.ibm.com/instances/d46cbead-4dd1-4a82-8370-f15b41b8ecc6", "x":410, "y":320, "wires":[[{"id":"5fa4960f.888e68", "a340b2de.770a7", "fd123896.430968", "f7bda789.c862a8", "c823fca.b8dd6", "1572dc84.0a5423", "9d4c7813.86c028"}]], {"id":"3bf056da.b636ba", "type":"ui_text", "z":"1b8e042d.9c8f6c", "group":"2f3a0665.46bd5a", "order":5, "width":"0", "height":"0", "name":""," "label":"Search
```

```
Output", "format":"{{msg.payload}}", "layout":"row-spread", "x":900, "y":120, "wires":[]], {"id":"f7bda789.c862a8", "type":"function", "z":"1b8e042d.9c8f6c", "name":"url", "func":"msg.payload=msg.search_results.results[0].url\nreturn
```

```
msg;", "outputs":1, "noerr":0, "x":630, "y":300, "wires":[[{"id":"332f40d4.9a2b"}]]
```

```
}}, {"id": "a340b2de.770a7", "type": "function", "z": "1b8e042d.9c8f6c", "name": "author", "func": "msg.payload=msg.search_results.results[0].author\n\nreturn\nmsg;", "outputs": 1, "noerr": 0, "x": 630, "y": 160, "wires": [[ "583b807b.65df3\n"] ]}, {"id": "1572dc84.0a5423", "type": "function", "z": "1b8e042d.9c8f6c", "name": "main page\nurl", "func": "msg.payload=msg.search_results.results[0].main_image_url\n\nreturn\nmsg;", "outputs": 1, "noerr": 0, "x": 640, "y": 240, "wires": [[ "6892e807.a3a77\n8" ] ]}, {"id": "fd123896.430968", "type": "function", "z": "1b8e042d.9c8f6c", "name": "score", "func": "msg.payload=msg.search_results.results[0].enriched_text.sentiment.document.score\n\nreturn\nmsg;", "outputs": 1, "noerr": 0, "x": 630, "y": 360, "wires": [[ "f9bfb192.33c39\n"] ]}, {"id": "c823fca.b8dd6", "type": "function", "z": "1b8e042d.9c8f6c", "name": "label", "func": "msg.payload=msg.search_results.results[0].enriched_text.sentiment.document.label\n\nreturn\nmsg;", "outputs": 1, "noerr": 0, "x": 610, "y": 420, "wires": [[ "564bff50.60b5b\n"] ]}, {"id": "564bff50.60b5b", "type": "ui_text", "z": "1b8e042d.9c8f6c", "group": "2f3a0665.46bd5a", "order": 7, "width": "0", "height": "0", "name": "", "label": "Label\n\n:", "format": "{ {msg.payload} }", "layout": "row-spread", "x": 850, "y": 400, "wires": []}, {"id": "332f40d4.9a2b", "type": "ui_text", "z": "1b8e042d.9c8f6c", "group": "2f3a0665.46bd5a", "order": 4, "width": "6", "height": "1", "name": "", "label": "URL
```

```
:", "format": "{{msg.payload}}", "layout": "row-spread", "x": 830, "y": 280, "wires": [], {"id": "583b807b.65df3", "type": "ui_text", "z": "1b8e042d.9c8f6c", "group": "2f3a0665.46bd5a", "order": 2, "width": "0", "height": "0", "name": "", "label": "Author of Article"}, {"id": "6892e807.a3a778", "type": "ui_text", "z": "1b8e042d.9c8f6c", "group": "2f3a0665.46bd5a", "order": 3, "width": "6", "height": "1", "name": "", "label": "Main Page URL"}, {"id": "f9bfb192.33c39", "type": "ui_gauge", "z": "1b8e042d.9c8f6c", "name": "", "group": "2f3a0665.46bd5a", "order": 6, "width": "0", "height": "0", "gtype": "gage", "title": "Sentiment score", "label": "units", "format": "{{value}}", "min": "-2", "max": "2", "colors": ["#00b500", "#e6e600", "#ca3838"], "seg1": "", "seg2": "", "x": 840, "y": 340, "wires": []}, {"id": "e7b1f4ed.5219a8", "type": "function", "z": "1b8e042d.9c8f6c", "name": "retrival", "func": "msg.payload=\n\n  {\n    \"series\": [\"News\"],\n    \"data\": [msg.payload[0], msg.payload[1], msg.payload[2]],\n    \"labels\": [\"Positive\", \"Negative\", \"Neutral\"]\n  }\n\nreturn msg;", "outputs": 1, "noerr": 0, "x": 790, "y": 540, "wires": [{"id": "736fc0db.4fa57", "type": "ui_chart", "z": "1b8e042d.9c8f6c", "name": "", "group": "2f3a0665.46bd5a", "order": 8, "width": "0", "height": "0", "label": "Garphical Representation"}, {"chartType": "bar", "legend": "false", "xformat": "HH:mm:ss", "interpolat
```



```
e":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":
1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOn
eColor":false,"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e",
"#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useO
ldStyle":false,"outputs":1,"x":1020,"y":480,"wires":[[[]]},{ "id":"9d4c781
3.86c028","type":"function","z":"1b8e042d.9c8f6c","name":"sentiment"
,"func":"var
val=[msg.search_results.aggregations[0].results[0].matching_results];\nv
al.push(msg.search_results.aggregations[0].results[1].matching_results);
\nval.push(msg.search_results.aggregations[0].results[2].matching_resul
ts);\nmsg.payload=val;\nreturn
msg;","outputs":1,"noerr":0,"x":640,"y":500,"wires":[["e7b1f4ed.5219a
8"]]}, {"id":"2f3a0665.46bd5a","type":"ui_group","z":"","name":"Using
WATSON","tab":"64857bf5.fd2e44","order":1,"disp":true,"width":"6","
collapse":false}, {"id":"64857bf5.fd2e44","type":"ui_tab","z":"","name":
"News search
APP","icon":"dashboard","order":2,"disabled":false,"hidden":false}]
```