

SMARTBRIDGE
REMOTE SUMMER INTERNSHIP
PROJECT REPORT
ON
AI POWERED NEWS SEARCH APP

BY
SOMA SHRENIKA

TABLE OF CONTENTS

1. INTRODUCTION
 - 1.1 Overview
 - 1.2 Purpose
 2. LITERATURE SURVEY
 - 2.1 Existing problem
 - 2.2 Proposed solution
 3. THEORITICAL ANALYSIS
 - 3.1 Block diagram
 - 3.2 Hardware / Software designing
 4. EXPERIMENTAL INVESTIGATIONS
 5. FLOWCHART
 6. RESULT
 7. ADVANTAGES & DISADVANTAGES
 8. APPLICATIONS
 9. CONCLUSION
 10. FUTURE SCOPE
 11. BIBILOGRAPHY
- APPENDIX
- A. Source code

1. INTRODUCTION

1.1 Overview

The era of mobile technology opens the window to the apps. Various applications have become part of our life. Here I am introducing 'AI POWERED NEWS SEARCH APP', it is an application which is used to fetch news for the user based on their interest. The system is a web mining application using Node-Red and IBM Watson Discovery Service. The server side application is built using Node Red. The user interacts with the app UI(Built with Node-RED) to request relevant news content. The app sends user requests to Watson Discovery News. The Watson Discovery Service is continually crawling the web to update its Discovery News collection. The Watson Discovery Service responds to Slack search requests. The user interface also shows the sentiment of a particular article on a sentimental gauge ranging from -20 to 20.

1.2 Purpose

The news is a part and parcel of regular life. If we want to stay updated with the regular updates, then it is vital to go through the daily news headlines and details. Once upon a time, radio, newspaper, and TV are the only modes of getting the daily news updates; however, after the inception of the digital world, people can get the updates in any place and anytime. With a news app, we can provide user-friendly navigation and better structured content that is personalized and diverse to make the users want to use the app more and more. As there are various sources of getting news, going through all the sources consumes the user time and they might end up reading the content of some topics only. When there is a provision for browsing the news the user requires, they can read the news of their interest. The sentiment analysis also helps the user in deciding whether to read an article or not ,based on the sentimental score.

2. LITERATURE SURVEY

2.1 Existing Problem

The newspapers take 24 hours of time to get published. Sometimes newspapers may not provide accurate global information, which can be found on web. A mobile application can be accessed in a much better way. An application provides a customized view of the news. The tone and context of the news play an important role which are neglected by many applications that provide news on web. Many of these problems are solved in this application.

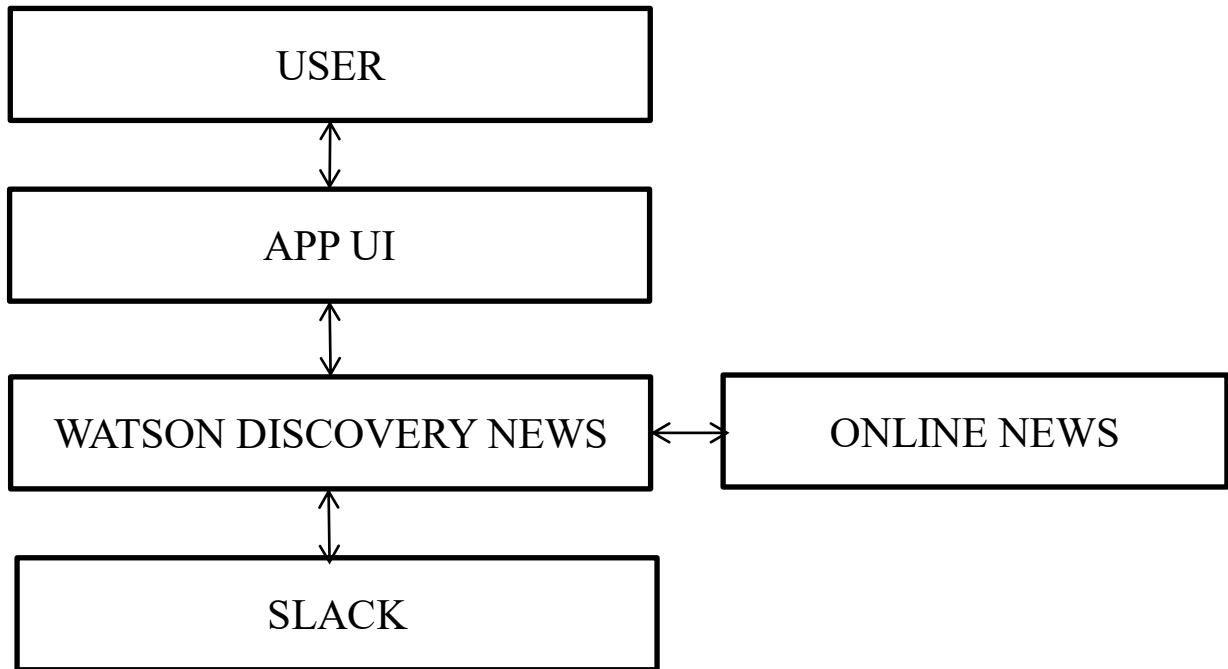
2.2 Proposed Solution

The solution is a news application that uses IBM services , slack and node red. These are intergrated to provide a UI that can be accessed by people. The integration is also extended by having a bot integration to a workspace on slack. The IBM services used are IBM Watson Discovery service, IBM node-red app. IBM Watson discovery consists of packages core Watson APIs such as Natural Language Understanding and Document Conversion along with UI tools that enable to easily upload, enrich, and index large collections of private or public data. IBM Watson Discovery News is included with Discovery. Watson Discovery News is an indexed dataset that is pre-enriched with the following cognitive insights: Keyword Extraction, Entity Extraction, Semantic Role Extraction, Sentiment Analysis, Relation Extraction, and Category Classification. Watson Discovery News is updated continuously with new articles, and is available in English, Spanish, German, Korean, French, and Japanese. Discovery News English is updated with approximately 300,000 new articles daily; Discovery News Spanish is updated with approximately 60,000 new articles daily; Discovery News German is updated with approximately 15,000 new articles daily; Discovery News Korean with 10,000 new articles daily; Discovery News French with 23,000 new articles daily; Discovery News Japanese is updated with approximately 17,000 new articles daily. The news sources vary by language, so the query results for each collection are not identical. 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.

3. THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Hardware / Software designing

1) APP UI

- The app UI is created by using node red on IBM cloud. The steps followed are
- Log in to IBM Cloud.
- Open the catalog and search for node-red.
- Click on the Software tab .
- Click on the Node-RED App tile.
- Click on the Create app button.
- Enable the Continuous Delivery feature.
- Open the node red application.
- Configure the application.

2) WATSON DISCOVERY NEWS

- Create an instance of the discovery service.
- Launch the tool.
- It consists of pre-enriched data.
- The api credentials are used for the integration with the node red app created earlier.
- The api key, url, collection ID, environment ID data is provided in the discovery node in the node red flow editor.
- This crawls the web all the time to fetch the result for the query given by the user.

3) SLACK

- The application is also integrated with slack as a bot to provide results for the information asked by the user.
- A slack bot is integrated to the current workspace.
- Navigate to the custom integrations sections and click on bots.
- Add a bot by clicking on add configuration.
- Enter the user name for the bot and save the integration.

Deployment

- The application is deployed locally by using node js app.
- The credentials of Watson discovery service and the api token from slack and placed in a .env file.
- Then the application is installed by using the command: `npm install`
- The application is run by using the command: `npm start`
- The bot starts to work after successfully deploying it.

4. EXPERIMENTAL INVESTIGATIONS

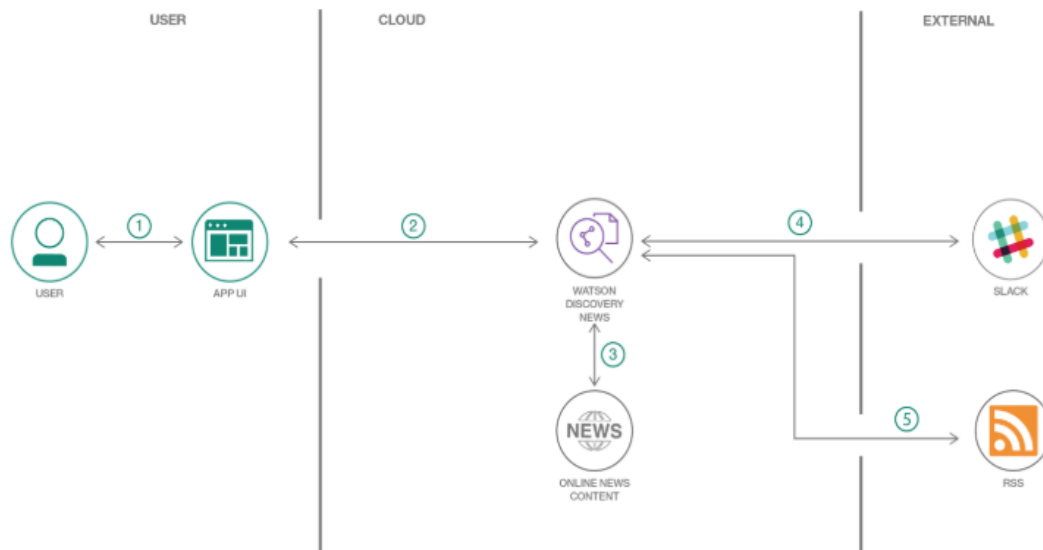
The application is build using JavaScript, Node.js, and the Watson Discovery Service.

- Code is written in Node.js, with the server-side using the Express framework and the client using ReactJS.
- The pre-built Watson Discovery News collection is used to find the news.
- Accessing the Watson Discovery Service is done through the Discovery API.

The app demonstrates

Search: Query for the most relevant new 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.

5. FLOWCHART



6 .RESULT

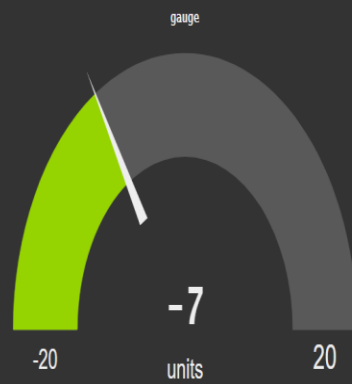
- The result of the application is the information about the query the user asked for.
- The results can be obtained either on the web by using the url for the UI.
- The results can also be obtained by using the slack bot in a slack workspace.

APP URL: <https://node-red-accpg.eu-gb.mybluemix.net/ui>

Default

Search result

In Video: COVID-19: No
changes in SSLC (Class
10) board exams in
Karnataka RELATED NEWS
India's 1st COVID-19
death in Karnataka Mar
13, 2020 India's first
COVID-19 death confirmed
in Karnataka Mar 12,
2020 Who's afraid of
COVID-19?Sentiment : -7




enter the query
covid

SUBMIT


CANCEL

shrenika-bot ☆

Messages
About




Soma Shrenika 12:59 PM
@shrenika-bot news please




shrenika-bot (APP) 1:00 PM
Hi there!


What news are you interested in?




Soma Shrenika 1:00 PM
AI



shrenika-bot (APP) 1:00 PM
You want me to search for news articles about AI?





Soma Shrenika 1:00 PM
yes




shrenika-bot (APP) 1:00 PM
OK searching...

<http://www.deadline.news.co.uk/2020/03/20/1139610/> (edited)


Deadline News
How does machine learning propel AI ops? - Deadline News
AI Ops is the new buzzing term in the world of IT and business. If you are a part of an IT organization, you must be hearing a lot about AI for IT ops and enterprise ai and how AI and Machine Learning can benefit your systems, processes, and productivity. This article focuses on AI
Mar 20th (107 kB)



Here are some news articles...
<https://www.kellogg.northwestern.edu/executive-education/individual-programs/online-programs/ol-aibt.aspx> (edited)


kellogg.northwestern.edu
Artificial Intelligence Strategies | Kellogg Executive Education

Message shrenika-bot

B I G L U

7. ADVANTAGES & DISADVANTAGES

Advantages

- Accurate data along with analytics.
- Improved customer interaction.
- Easy provision of information.
- Easy navigation.

Disadvantages

- Technical issues.

- During the maintenance of the services customers will have only limited access.

8. APPLICATIONS

- They can be integrated with various platforms to provide more functionality and provide information to stay updated.
- Increase the level of accessibility.
- Provide specific articles.

9. CONCLUSION

The application developed is an intelligent news search app. It uses AI to find the results. The application is developed by using IBM cloud services and IBM node red app and integrating it with slack and also doing sentimental analysis on the results.

10. FUTURE SCOPE

- In future the app can be integrated with other popular applications.
- The number of results displayed can also be specified by the user.
- Voice search can also be provided.

11. BIBLIOGRAPHY

- <https://developer.ibm.com/tutorials/how-to-create-a-node-red-starter-application/>
- <https://nodered.org/>
- <https://github.com/watson-developer-cloud/node-red-labs>
- <https://www.ibm.com/watson/products-services>
- <https://www.youtube.com/watch?v=W3iPbFTAAAds&feature=youtu.be>
- <https://developer.ibm.com/articles/introduction-watson-discovery/>
- <https://cloud.ibm.com/docs/services/discovery?topic=discovery-getting-started>
- <https://www.youtube.com/watch?v=kwmqJRDbv98&feature=youtu.be>
- <https://cloud.ibm.com/docs/discovery?topic=discovery-query-concepts#querying-news>
- <https://discovery-news-demo.ng.bluemix.net/>
- <https://github.com/IBM/watson-discovery-news/>

APPENDIX

A. Source code

App.js

```
require('dotenv').config({ silent: true });

const server = require('./server');
const port = process.env.PORT || process.env.VCAP_APP_PORT || 3000;

server.then(app => {
  app.listen(port, () => {
    require('./server/slack-bot');
    // eslint-disable-next-line no-console
    console.log('Watson Discovery News Server running on port: %d', port);
  });
});
```

Query-builder.js

```
const util = require('util');
const moment = require('moment');
const aggregations = {
  search: 'term(enriched_text.sentiment.document.label)',
  trending: 'term(enriched_title.entities.text,count:20).top_hits(1)'
};

module.exports = {
  aggregations,
  setEnvironmentId(environmentId) {
    this.environment_id = environmentId;
  },
  setCollectionId(collectionId) {
    this.collection_id = collectionId;
  },
  search(queryOpts) {
    const params = Object.assign({
      environment_id: this.environment_id,
      collection_id: this.collection_id,
      count: 10,
      // sort: 'result_metadata.score',
    });
```

```

        return: 'enriched_text.sentiment.document.label',
        aggregation: aggregations.search
    }, queryOpts);

    console.log('Discovery Search Query Params: ');
    console.log(util.inspect(params, false, null));

    return params;
},
trending(queryOpts = {}) {
    const { filter } = queryOpts;
    const timeAndSourceFilter =
`crawl_date>${moment().subtract(24,'h').toISOString().slice(0, -5)}`;

    const params = Object.assign({
        environment_id: this.environment_id,
        collection_id: this.collection_id,
        return:
'title,text,url,host,crawl_date,result_metadata.score,id,enriched_text.entities.text
,enriched_text.sentiment.document.label',
        aggregation: aggregations.trending
    }, queryOpts, {
        filter: filter ? `${filter},${timeAndSourceFilter}` : timeAndSourceFilter
    });

    console.log('Discovery Trending Query Params: ');
    console.log(util.inspect(params, false, null));
    return params;
}
};

```

Slack-bot.js

This is a sample Slack bot built with Botkit.

This bot demonstrates many of the core features of Botkit:

- * Connect to Slack using the real time API
- * Receive messages based on "spoken" patterns
- * Reply to messages

- * Use the conversation system to ask questions
- * Use the built in storage system to store and retrieve information for a search queries.
- * Search for news using the Watson Discovery API by hitting the api endpoint of the server.

RUN THE BOT:

Get a Bot token from Slack:

-> <http://my.slack.com/services/new/bot>

USE THE BOT:

Find your bot inside Slack to send it a direct message.

Say: "Hello"

The bot will reply "Hello!"

Say: "news please" or "whats in the news"

The bot will ask you What news are you interested in?

Say: "merger and acquisition of artificial intelligence companies"

Confirm and the bot will show you links to top three news articles from Watson Discovery News Service

Make sure to invite your bot into other channels using /invite @<my bot>!

~~~~~  
~~~~~\*/

```
require('isomorphic-fetch');
const queryString = require('query-string');
const port = process.env.PORT || process.env.VCAP_APP_PORT || 3000;
const slackBotToken = process.env.SLACK_BOT_TOKEN;
```

```
const Botkit = require('botkit');
const controller = Botkit.slackbot();
```

```

if (!slackBotToken) {
  // eslint-disable-next-line no-console
  console.log('Warning: SLACK_BOT_TOKEN not specified so functionality
will be disabled');
}

// eslint-disable-next-line no-unused-vars
const bot = controller.spawn({
  token: slackBotToken
}).startRTM();

controller.hears(['hello', 'hi'], 'direct_message,direct_mention,mention',
function(bot, message) {

  bot.api.reactions.add({
    timestamp: message.ts,
    channel: message.channel,
    name: 'robot_face',
  }, function(err) {
    if (err) {
      bot.botkit.log('Failed to add emoji reaction :(', err);
    }
  });

  bot.reply(message, 'Hello.');
```

```

});

controller.hears(['whats in the news', 'news please'],
'direct_message,direct_mention,mention',
function(bot, message) {
  bot.startConversation(message, function(err, convo) {
    if (!err) {
      convo.say('Hi there!');
      convo.ask('What news are you interested in?', function(response, convo) {
        convo.ask('You want me to search for news articles about \'' +
response.text + '\'', [
      {
        pattern: bot.utterances.yes,
        callback: function(response, convo) {
```

```

        // since no further messages are queued after this,
        // the conversation will end naturally with status == 'completed'
        convo.next();
    }
},
{
    pattern: bot.utterances.no,
    callback: function(response, convo) {
        // stop the conversation. this will cause it to end with status ==
'stopped'
        convo.stop();
    }
},
{
    default: true,
    callback: function(response, convo) {
        convo.repeat();
        convo.next();
    }
}
]);

```

```

    convo.next();

}, {'key': 'search-query'});

convo.on('end', function(convo) {
    if (convo.status == 'completed') {
        bot.reply(message, 'OK searching...');

        const qs = queryString.stringify({ query:
convo.extractResponse('search-query') });
        const host = `http://localhost:${port}`;
        // eslint-disable-next-line no-console
        console.log(`Slack Bot host route: ${host}`);
        fetch(`${host}/search/api/search?${qs}`)
            .then(apiResponse => {
                if (apiResponse.ok) {
                    apiResponse.json()
                        .then(json => {

```

```

        bot.reply(message, 'Here are some news articles...');
        for (let i = 0; i < 3; i++) {
            setTimeout(() => {
                bot.reply(message, `<${json.results[i].url}>`);
            }, i * 1000);
        }
    });
} else {
    throw new Error(apiResponse.json());
}
})
.catch(err => {
    // eslint-disable-next-line no-console
    console.error('error', err);
    bot.reply(message, 'Error fetching news');
});
} else {
    // this happens if the conversation ended prematurely for some reason
    bot.reply(message, 'OK, nevermind!');
}
});
}
});
});

```

```

controller.hears(['identify yourself', 'who are you', 'what is your name'],
    'direct_message,direct_mention,mention', function(bot, message) {
        bot.reply(message, ':robot_face: I am a bot named <@' + bot.identity.name +
        '>');
    });

```