CAB432 CLOUD COMPUTING

News Content Analysis: A Web Mashup

Quang Huy Tran | n10069275

Table of Contents

Introduction

News Content Analysis is a web mashup developed was part of the course requirements for CAB432, Semester 2, 2019. News Content Analysis aims to provide users with a summary of content analysis from top headlines in four different countries.

Users can view the content labels of top headlines in their chosen country with utilising News and Cloud Natural Language APIs. The relevant results are presented in short articles with title, description, images and published date alongside the predicted labels. From this point, the user can click on a given button on any article to view all news that have similar labels. For a visual representation of full user guide, please see Appendix A.

News API

Home URL: https://newsapi.org

News API is a simple HTTP REST that allows the user to search news outlets all around the world. These articles have been tracked in 7 categories across over 50 countries, and at over a hundred top publications and blogs, in near real-time. News API has three endpoints:

- Top headlines (/v2/top-headlines): this endpoint returns breakings news headlines for a country and category, or from a single or multiple sources.
- Everything (/v2/everything): this endpoint allows the user to search every recent year recent news and blog article published by 30,000 different sources large and small.
- Sources (/v2/sources): this endpoint returns information (including name, description, and category) about the most notable sources.

Cloud Natural Language API (Google)

Home URL: https://cloud.google.com/natural-language/

Natural Language uses machine learning to reveal the structure and meaning of text. The powerful pre-trained models of the Google Natural Language API let developers work with natural language understanding features including sentiment analysis, entity analysis, entity sentiment analysis, content classification, and syntax analysis. The Natural Language API is a REST API, and consists of JSON requests and responses using one of Google Cloud Client Libraries (including Python, Node.js, Java, Go, PHP, C# and Ruby).

Mashup Use Cases and Services

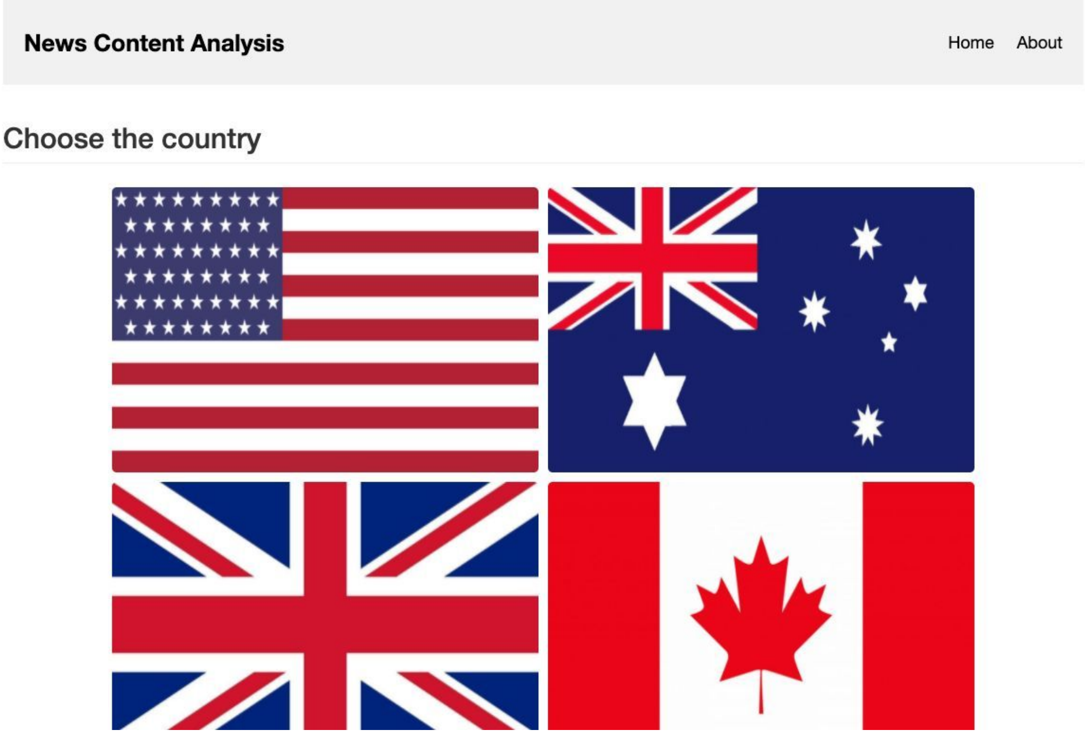
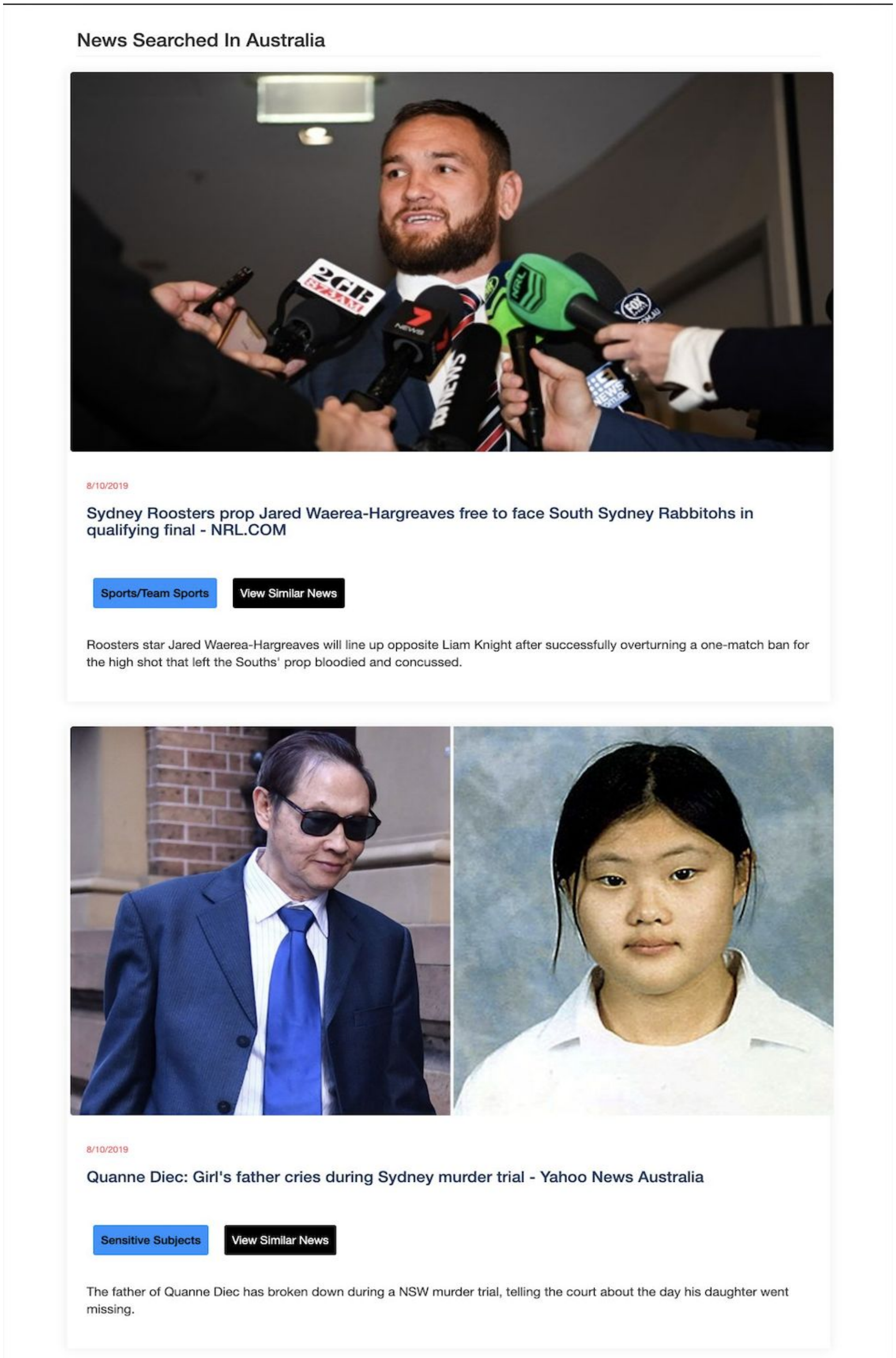


Figure 1: Homepage website

Use Case 1

“As a news reader, I want to see a list of top headlines based on my selected country with a content label attached, so that I can select quickly the kind of article I would like to read”.



News Searched In Australia

8/10/2019

Sydney Roosters prop Jared Waerea-Hargreaves free to face South Sydney Rabbitohs in qualifying final - NRL.COM

Roosters star Jared Waerea-Hargreaves will line up opposite Liam Knight after successfully overturning a one-match ban for the high shot that left the Souths' prop bloodied and concussed.

8/10/2019

Quanne Diec: Girl's father cries during Sydney murder trial - Yahoo News Australia

The father of Quanne Diec has broken down during a NSW murder trial, telling the court about the day his daughter went missing.

Figure 2: The results of selecting the source news in Australia

Figure 1 represents the home page website of News Content Analysis application. The users have the option of selecting one country out of four countries (USA, Australia, UK and Canada). Then, Figure 2 illustrates the results of how top articles are being displayed after selecting the source news in Australia. Each article is attached a label which is predicted based on the article content. The user can click on the title to see the article in the original source.

This use case relies on two APIs:

News API

In this use case, the chosen country will be passed to News API endpoint (https://newsapi.org/v2/top-headlines?country=?&apiKey=API_KEY) as a parameter request. This endpoint will return JSON object containing top headlines from the chosen country.

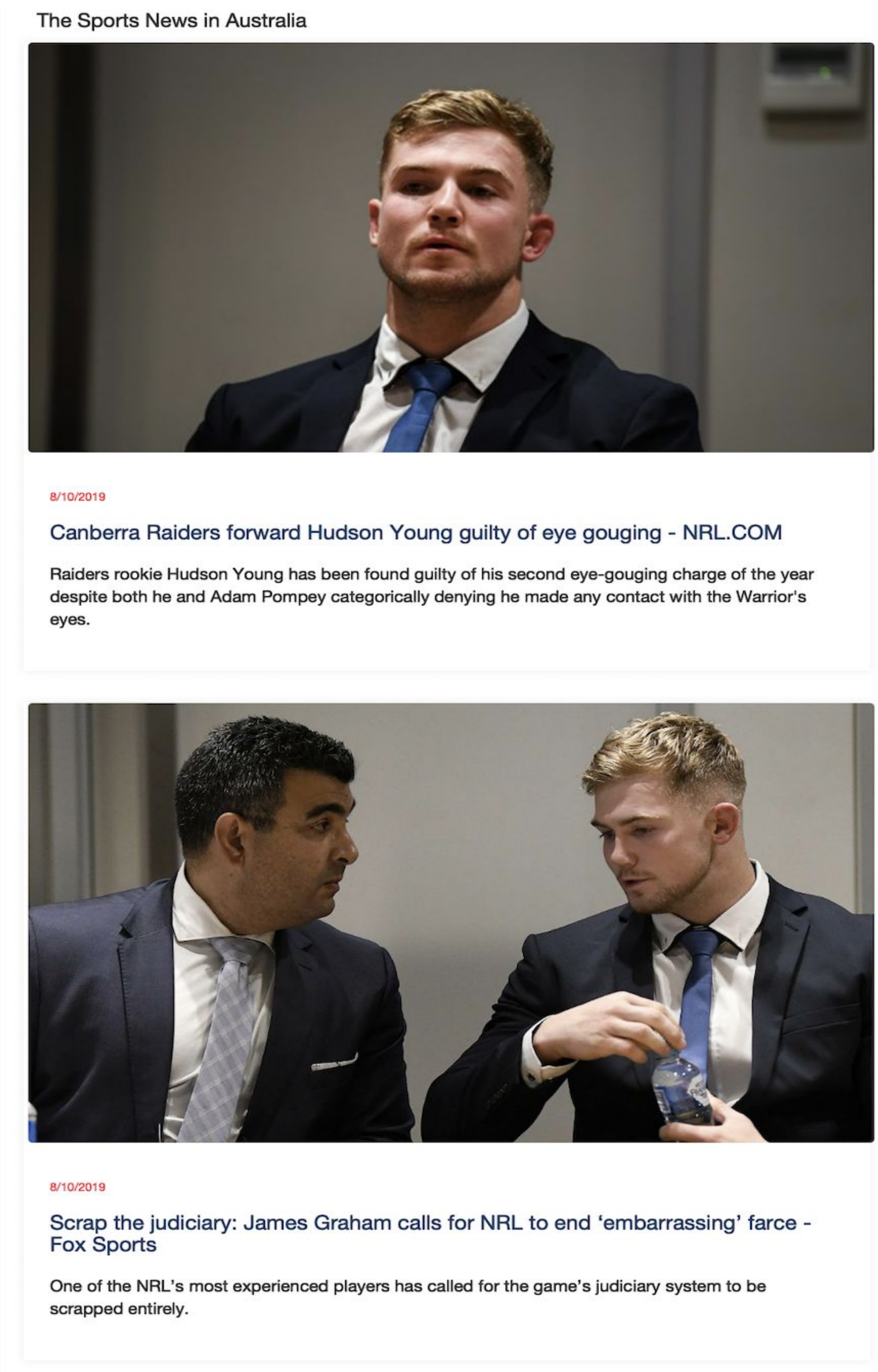
Cloud Natural Language API

Once the JSON object has been retrieved completely from the endpoint, each article from response object will be passed to a JavaScript function that is used to filter, analyse the article and predict the content label with a confidence score.

Use Case 2

“As a news reader, after viewing an article that I am interested in, I want to find other articles within the content label of this article”.

It is observed from Figure 2 that each headline includes a content label in blue color and a button “View Similar News” in black color. When a user press this button, it will redirect them to another page that presents a list of all articles in the same category as the content label. For example, the first article in Figure 2 has a “Sports/Team Sports” tag, pressing the button will show the results of all sports articles in Australia as illustrated in Figure 3.



The Sports News in Australia

8/10/2019

Canberra Raiders forward Hudson Young guilty of eye gouging - NRL.COM

Raiders rookie Hudson Young has been found guilty of his second eye-gouging charge of the year despite both he and Adam Pompey categorically denying he made any contact with the Warrior's eyes.

8/10/2019

Scrap the judiciary: James Graham calls for NRL to end 'embarrassing' farce - Fox Sports

One of the NRL's most experienced players has called for the game's judiciary system to be scrapped entirely.

Figure 3: The articles with sports category

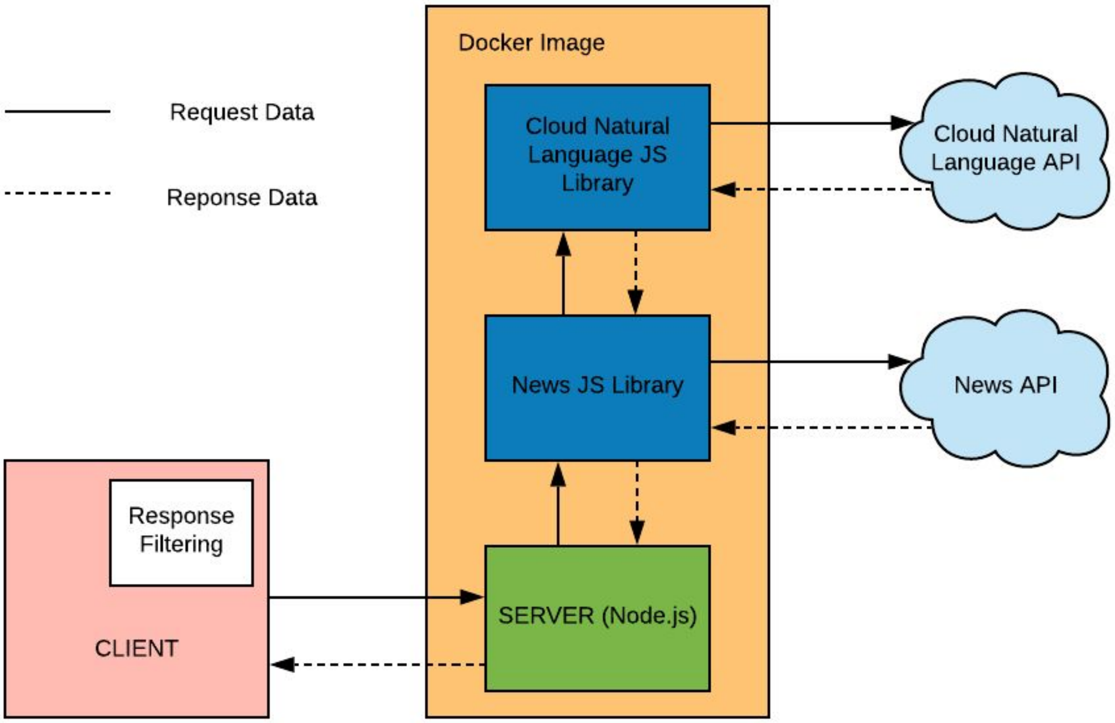
This use case relies on an API:

News API

In this use case, the content label predicted by Cloud Natural Language API will be transformed into categories of News API by if else conditions in an EJS template. Then, the category and country parameters will be passed to News API endpoint (https://newsapi.org/v2/top-headlines?country=?&category=?&apiKey=API_KEY) to get all related headlines and then displayed on client-side web page.

Technical Description

A basic overall representation of the flow of data can be seen in the figure below



Client

Initially, the client will initiate a request to the server, in which they will be served the index page. At this point, a request will be formed based on the country selected by the user and it will be sent to the server via a GET request.

Given the response by the server, the relevant information is processed and displayed accordingly. In each article, clicking a "View Similar News" button will make a request to the server via a GET request with country and predicted category parameters. A response filtering part implemented in client is used to transform the categories of Natural Language API into those of News API by if else conditions for 2nd use case.

The client-side technologies are:

- HTML5
- CSS3
- JavaScript

Using EJS (Embedded JavasScript Templates) as the view engine

Server

Given a request from the client, the server deals with the extended requests to externals APIs and responds to the client with the relevant information. The Node server deals solely with routing of incoming client requests and outgoing responses, with additional JS functions for predicting the content labels and handling requests to external APIs

The web server for this mashup runs the ExpressJS web framework, built on Node.js. This service utilises several node packages, the most important of which are outlined below:

- express - this is the web framework on which the service is built
- ejs - a simple templating language that lets you generate HTML markup with plain JavaScript
- express-ejs-layouts - layout support for ejs in express
- body-parser - a middleware package that allows parsing of response body when making HTTP requests.
- newsapi - a node interface for NewsAPI
- dotenv - storing configuration in the environment separate from code is based on The Twelve-Factor App methodology

- @google-cloud/language - a node interface for Cloud Natural Language API
- nodemon - a tool that helps develop node.js based applications by automatically restarting the node application when file changes in the directory are detected

Project Structure

The project structure is a general Express Node.js web application that is great for separating the responsibility of the different parts of the app and makes the code easier to maintain. A brief explanation of what each of them is about:

- routes/ - defines the app routes and their logic
- helpers/ - contains helper functions
- public/ - contains css files
- views/ - provides EJS templates which are rendered and served by the routes folder
- server.js - initialize the app and glue everything together
- package.json - remember all packages that the app depends on and their versions

Issues

The issue identified early on was the extensive nesting callbacks in HTTP GET requests, which became a particular problem when trying to iterate over all the News API results and perform a content prediction for each article. This problem was solved by using the Promise.all() function that was proved to be a particularly useful and concise solution to the aforementioned problem.

Docker Discussion

The Docker configuration for this application is quite straightforward. The base image is node:10, the file JSON obtained from Cloud Natural Language API is then exported as an environment variable and then file package.json is copied into the directory to install necessary dependencies. After installing, the application directory is copied and the port on which node server runs is then exposed (8000). Finally, the Dockerfile uses CMD ‘npm start’ to start the application, making use of the *start* command found in News

Content Analsysis's *package.json* file, 'node server.js', to run the Node server. The Dockerfile can be found in Appendix B.

Testing and Limitations

Error Handling

The API response errors in predicting the content labels are caught through try-catch method. Additionally, the errors in formatting the requests are caught by analysing the status response and then redirecting to home page with an appropriate error message.

Test Cases

Step	Expected Outcome	Result
Access port 8000 of local network	Application loads home page	PASS
Click on "About" button at the upper right side of the home page	Application loads the /about, showing the purpose of the application	PASS
Click on "Home" button at the upper right side of the home page	Application loads home page	PASS
Click on a country flag at once	Application loads the /search/{country}, displaying the list of articles with images, title and content labels	PASS
Choose a specific article, press "View Similar News" button	Application loads the /similar/{country}/{category}, displaying all articles in the same category in a chosen country	PASS
Load an error request in the route /search/vietnam	Application returns to home page with an error message	PASS

Load an error request in the route /similar/vietnam/healthy	Application returns to home page with an error message	PASS

Compromises

During testing, it is noticed that there are differences in the list of categories between two APIs used in this project. The list of content categories of Cloud Natural Language API are very detailed with more than 100 results, however News API only allows us to access 7 different categories (business, technology, science, health, entertainment, sports and general). This gap of categories affected on some results of the second use case as converting the category from Google into News API is not exactly correct.

Aside from these user tests, tests were run on accessing the web routes, including forcing errors and checking how they are displayed. At present, there are no major failings of the application.

Possible Extensions

In the future, this service could be extended in several ways. Firstly, it would be good to add integration with Google Translate API, in order to display the top headlines in different languages across many countries. Furthermore, this service could be potentially be extended to analyse the entities in the content of each article and then allow the user to view articles with the same keyword. Finally, the service could potentially be extended to include statistics of predicted content labels, most common words and sentiment analysis from top news across all countries.

References

News API Documentation. (2018). Retrieved September 12, 2019, from https://newsapi.org/docs

Cloud Natural Language API Documentation (2016). Retrieved September 12, 2019, from https://cloud.google.com/natural-language/docs/

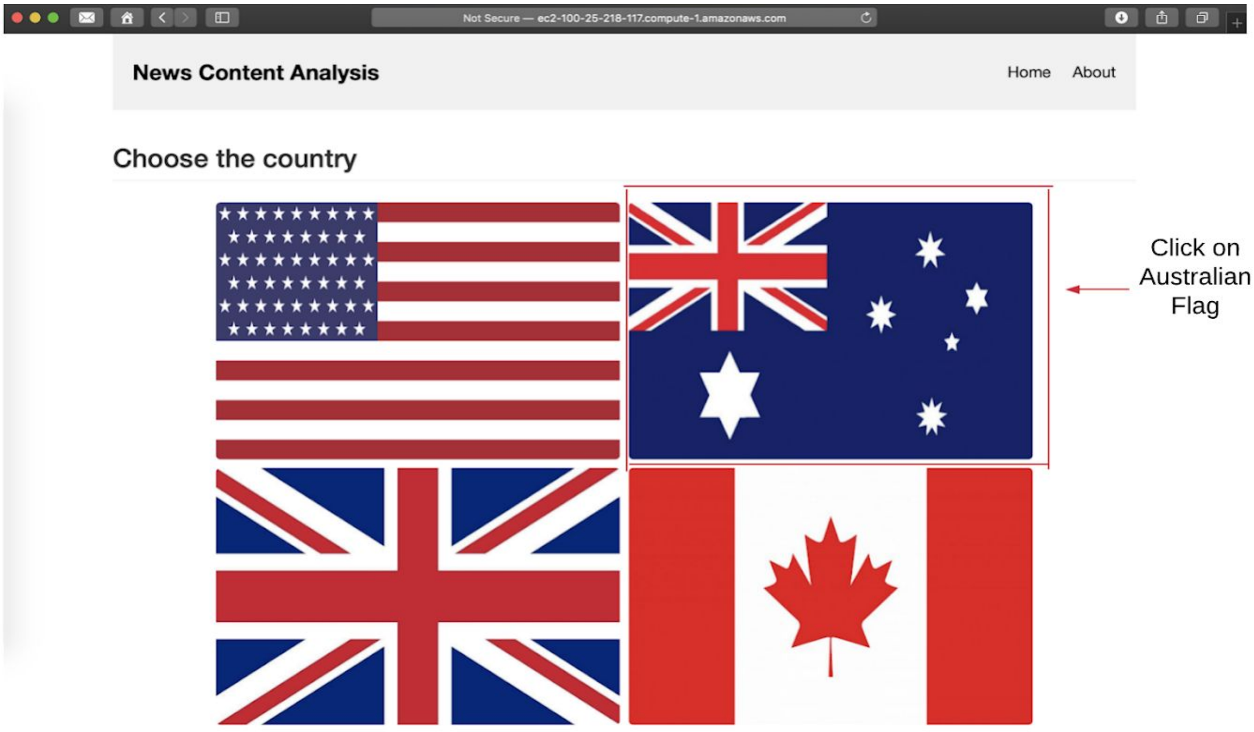
Fidanov, S. Best practices for Express app structure. Retrieved September 12, 2019, from https://www.terlici.com/2014/08/25/best-practices-express-structure.html

Appendix

Appendix A: User Guide (Australian News)

Use Case 1

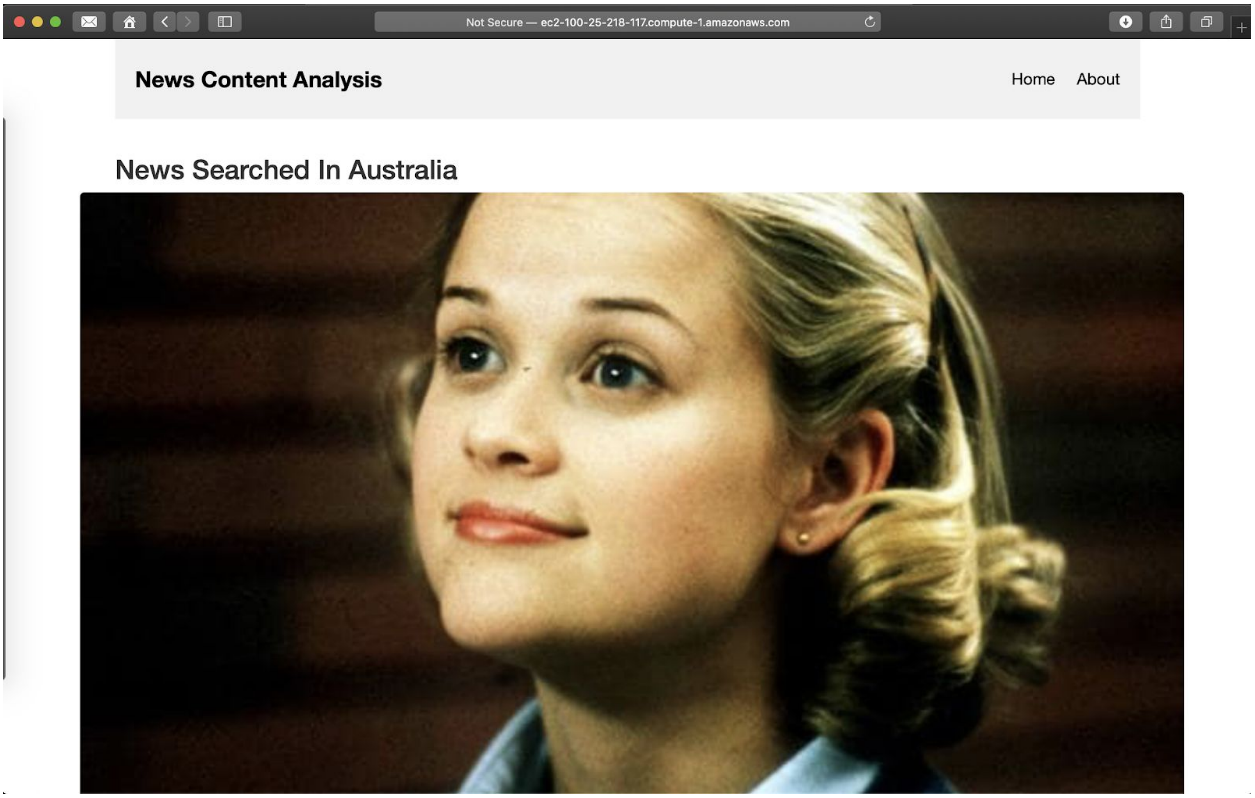
Load Website and Select Country (Australia)



News Content Analysis

Choose the country

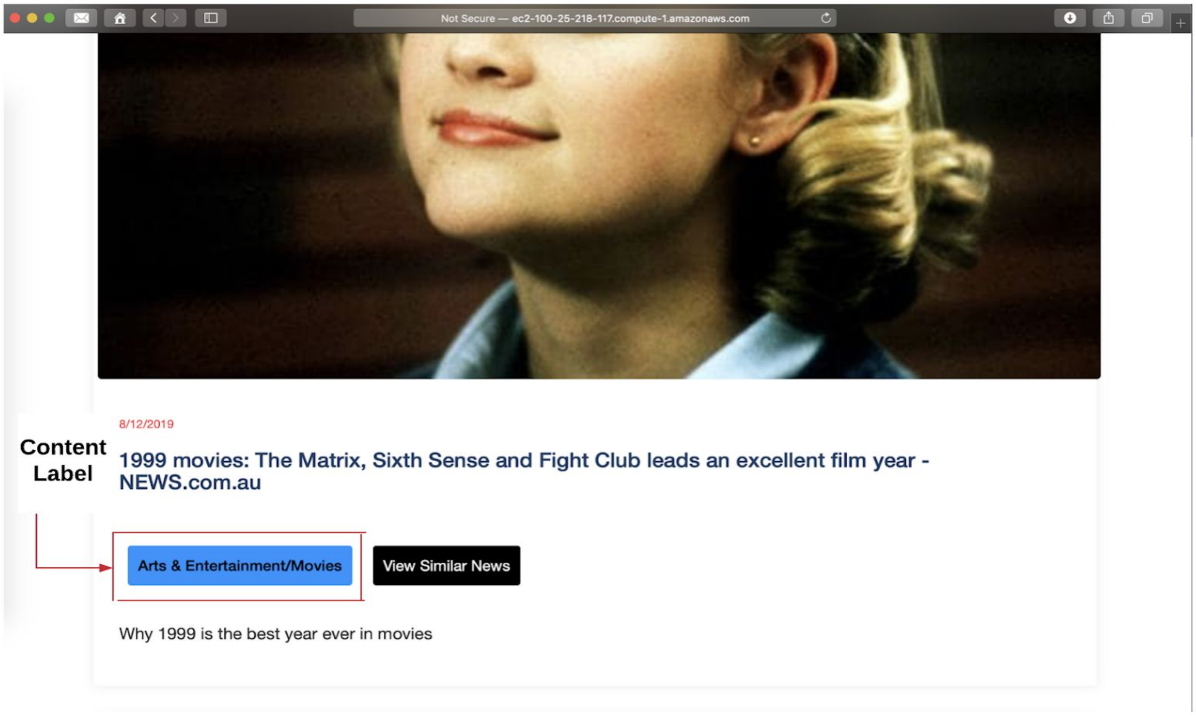
Click on Australian Flag



News Content Analysis

News Searched In Australia

View the article title and its content label


Not Secure — ec2-100-25-218-117.compute-1.amazonaws.com
Content Label
Arts & Entertainment/Movies
View Similar News

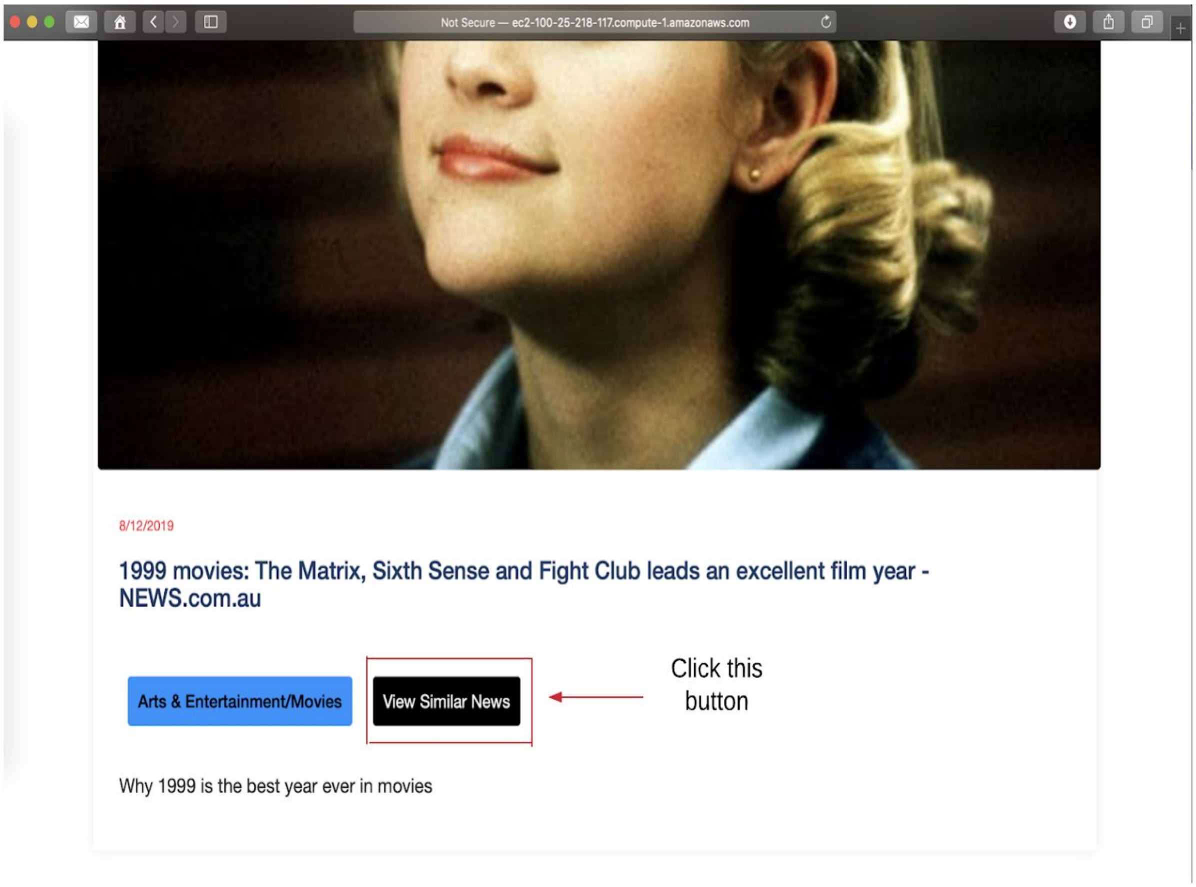
8/12/2019

1999 movies: The Matrix, Sixth Sense and Fight Club leads an excellent film year - NEWS.com.au

Why 1999 is the best year ever in movies

Use Case 2

Click “View Similar News” button on article above

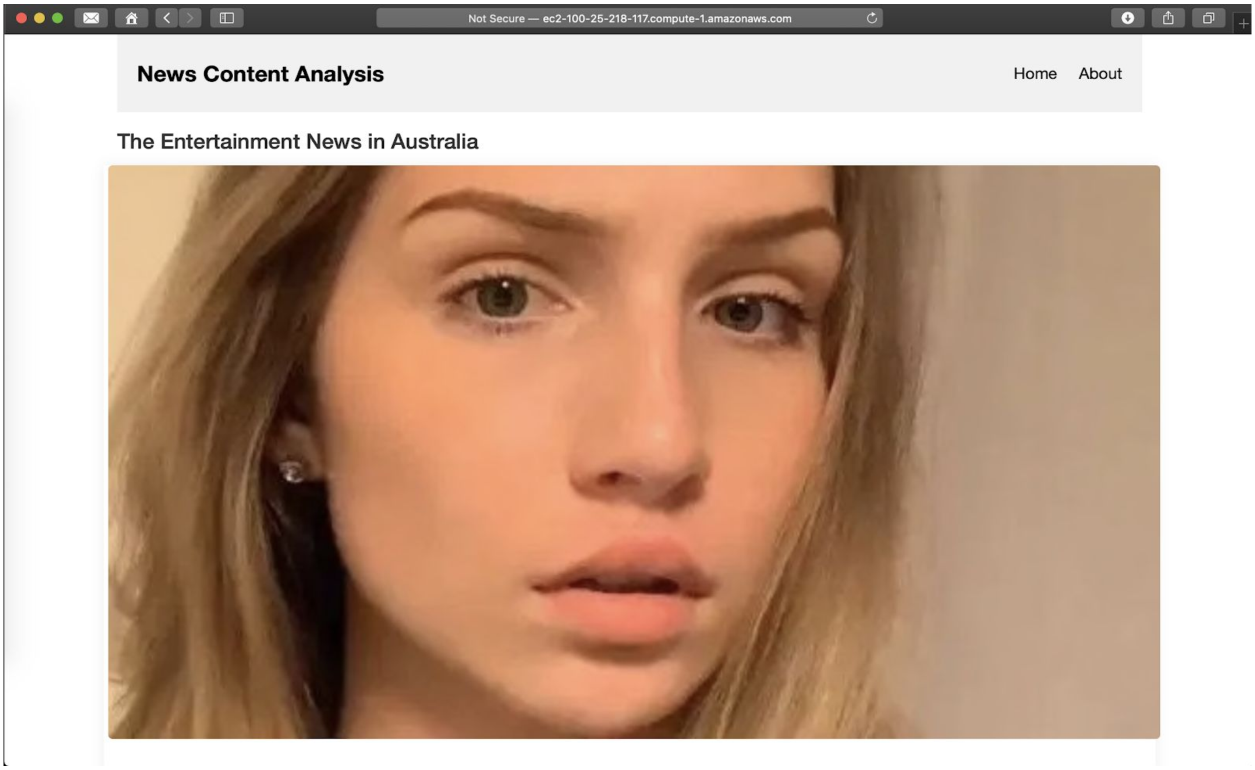

Not Secure — ec2-100-25-218-117.compute-1.amazonaws.com
Arts & Entertainment/Movies
View Similar News
Click this button

8/12/2019

1999 movies: The Matrix, Sixth Sense and Fight Club leads an excellent film year - NEWS.com.au

Why 1999 is the best year ever in movies

And the result:



News Content Analysis

Home About

The Entertainment News in Australia

Appendix B: Dockerfile

```
FROM node:10

RUN mkdir -p /src

WORKDIR /src

COPY app/huytran0801-af107e0998c5.json ./

ENV GOOGLE_APPLICATION_CREDENTIALS  huytran0801-af107e0998c5.json

RUN export GOOGLE_APPLICATION_CREDENTIALS=GOOGLE_APPLICATION_CREDENTIALS

COPY app/package*.json ./

RUN npm install

COPY /app .
```

```
EXPOSE 8000

CMD ["npm","start"]
```