Fake News Authenticity RESTful Service

Karthik Hubli, 01689129

08-May-2018

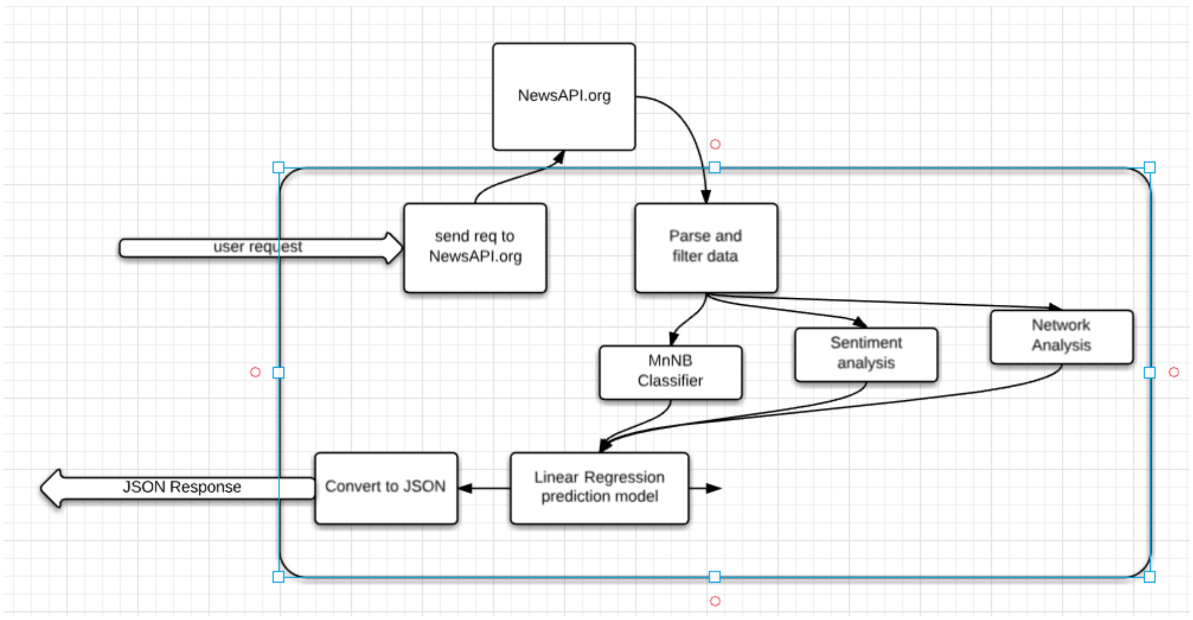
Table of Contents

1. Introduction

The application described below is implementation of the research done as part of directed research under Prof. Haim Levkowitz in the Spring 2018 semester. The findings of the research are outlined in detail in the research paper submitted along with this report. The report primarily outlines the implementation of research experiment and the outcome from it. The report discusses the choice of data, technology and the services used for implementation of the application. The application provides a service where the user can request for latest news article along with details about the analysis of the content. Machine learning and Natural Language Techniques are employed for analysis. The Machine Learning and NLP techniques used in the project are discussed in detail in the research paper submitted along with the report [*Predicting authenticity of news article using NLP and Machine Learning*]

2. Analysis and Design

The data is fetched from external source, “NewsAPI.org’ and the analysis is carried out on the ‘gunicorn’ webserver and delivered to the user in the form of JSON. The below diagram describes the high level architecture of the application.



3. Technology Stack

The primary programing language used for the application is Python 3.2.7. Python was chosen as it provides several libraries required for training and predicating using machine learning technique

	Requirement	Technology/library
1	Word Vectorization	TfidfVectorizer [sklearn]
2	Multinomial naive Bayes classifier	MultinomialNB [sklearn]
3	Sentiment Analysis	TextBlob
4	Webcscarping and parsing	Beautifulsoup [urllib]

5	Data loading and manipulation	Pandas, numpy
6	RESTful framework	Flask RESTplus
7	Web container	Heroku dyno
8	Web server	gunicorn
9	Version control	github

4. API

swaggerUI url: https://news-authenticity.herokuapp.com/api/documentation#/default

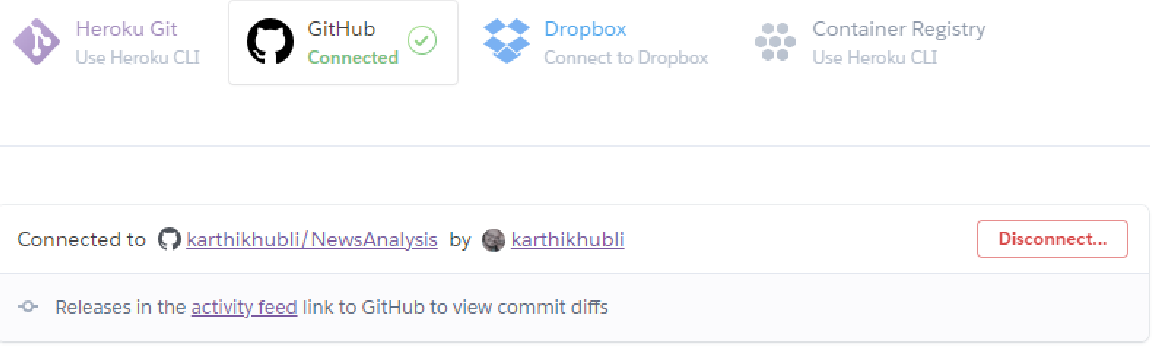
credibility	`https://news-authenticity.herokuapp.com/api/credibility?news=<body>&title=<title>`
headlines	`https://news-authenticity.herokuapp.com/api/headlines?apiKey=<api_key>`
newsbytopic	`https://news-authenticity.herokuapp.com/api/newsbytopic?apiKey=<api_key>&topic=<key_word>s&category=<category>`

5. Deployment of the System

The application is deployed on Heroku cloud service. It is deployed on free dyno of Heroku web container. The application is deployed on top the 'gunicorn' server.

The dyno is linked to the git hub repository. To update the application check in code in repository and deploy the same as below.

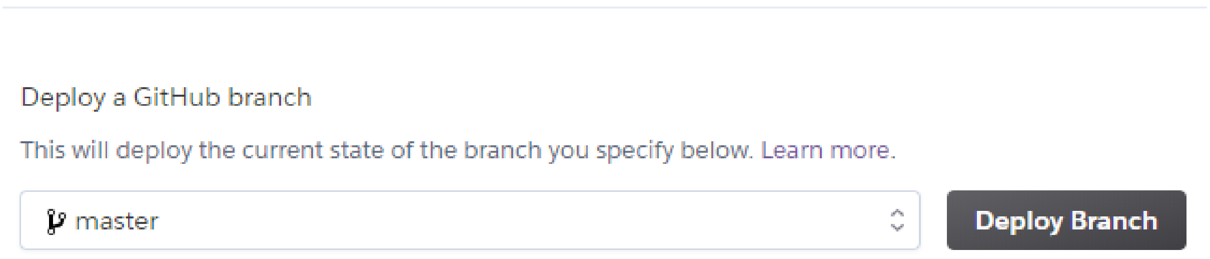
- Make sure the repository is liked with the dyno and necessary permissions are granted.



Connected to karthikhubli/NewsAnalysis by karthikhubli

Releases in the activity feed link to GitHub to view commit diffs

- Click on the 'Deploy Branch' and the dyno is updated automatically. Make sure the appropriate branch is slecated.


Deploy a GitHub branch

This will deploy the current state of the branch you specify below. Learn more.

6. Limitation of the System

The application is a backend service and it heavily relies on proper implementation of front end service. The application can be adapted either as desktop or mobile application. For live data, the application relies on external news aggregator thus there is no control over the data provided.

The application processes all the data on request and hence the render time is slow for a modern web application. Additionally, the analysis doesn’t categorize the news articles by the genre and the techniques employed for analysis is same for all the categories.

7. Conclusion and Future Work

The application provides a good spring board for a promising application. With few enhancements and addition to the infrastructure, it can be transformed into a sustainable and very effective product. The impact on users with such an application is large and can be very effective in combating lies.

In the future, I would be expanding the network analysis for the reference both ways. Currently, the application scans only if the application references any unreliable source. I would be adding additional feature to look for the article being referred by fake and propaganda news domains.

To improve the performance, I will be changing the architecture to two-stage system. Data aggregation, analysis and prediction will be handled by the existing python server application. This will be changed from a RESTful service to a batch job which gathers information at regular intervals and update the data to a NoSQL or a DocumentDB. The second stage will be a Nodejs based RESTful service which will query the Database and provide the content on need basis. The response will be lot faster as the content is pre-processed and ready to be consumed.

8. Reference

[1] Rodriguez, Alex. *Restful web services: The basics (2008)*
[2] Cholia, Shreyas and Skinner, David and Boverhof, Joshua. *NEWT: A RESTful service for building High Performance Computing web applications* (2010)
[3] Skirpan, Michael and Yeh, Tom. *Beyond the flipped classroom: learning by doing through challenges and Hack-a-thons* (2015)
[4] Bray, Tim. *The javascript object notation (json) data interchange format* (2017)
[5] Miguel Grinberg *https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask*
[6] John Kagga *https://medium.com/the-andela-way/deploying-a-python-flask-app-to-heroku-41250bda27d0*