The Reverse Prism

Understanding the global perception of news articles

Report By - **The Starks**
Akshaya Balamurugan (A0178458L)
Gananathan Khoteeswarun (A0178328U)
Monisha Prasad (A0178265U)
Venkateswara Venkata Krishnan (A0178343Y)

Table of Contents

I. Introduction

Waiting for the paper boy to deliver newspaper on a Sunday morning at 7 AM and gazing through the news with a sip of coffee, to getting notifications in the mobile device right at the break of any incident, times have changed in the way news is conveyed. Online news is now gaining higher momentum compared to the telecommunication medium owing to its perks of grabbing news on the go. Thus, every traditional newspaper company has established its online presence. 2018 is considered as the year when online media overtakes the traditional media [1]. Increasing information over the web always has its pros and cons. The reason for the trend shifts towards online medium is the accessibility and the availability of the news. It is on the hands of the online news providers to maintain the journalistic ethic. Two things which plays key role in the online news industry is how the news is delivered and how the news is perceived. This project aims to understand how a global event is captured and delivered in different parts of the globe. In this project, we handpicked 20 different news websites from 11 different regions across the world, searched for the news articles corresponding to a specific topic by considering the search made earlier for the same topic. Upon finding the articles, they are scraped for the header, content and the time of the article. The scrapped content is processed for insights discovery by summarizing, by extracting keywords and by sentiment mining. The derived insights are then better visualized using tableau.

II. Data Source

The data is extracted from the credible online news websites of various regions. Table 1 shows the list of regions and the news websites corresponding to those regions.

Region	Online News URLs
ANZ	https://www.news.com.au/ https://www.nzherald.co.nz/
India	https://www.thehindu.com/ https://indianexpress.com/ https://www.deccanchronicle.com/
Singapore	https://www.straitstimes.com/ https://www.channelnewsasia.com/
China	http://www.xinhuanet.com/english/ http://www.chinadaily.com.cn/
Russia	https://russia-insider.com/en
Europe	http://www.dailymail.co.uk https://www.neweurope.eu/ http://www.euronews.com/
US	https://www.usatoday.com/ https://www.nytimes.com/
Nigeria	https://www.vanguardngr.com/ http://dailypost.ng/
South Africa	https://www.iol.co.za/ https://mg.co.za/
UAE	https://gulfnews.com/
Mexico	https://mexicodaily.com

Table 1 – Various Data Sources

From these websites, the articles are scraped in a sequential process. After manually investigating each website and various articles from the websites, a generic structure for the news articles is obtained to enable the crawling process as a collective approach. The next section briefs about the common web page structure of the articles.

III. Structure of news articles

Figure A depicts the structure of a typical news article and the nature of HTML tags used in them. This is a collective generalized representation of all the 20 news websites crawled for this project. Apart from the structure below, there are few exceptions which were handled by scrutinizing individual sites.

Container - Wrapper for the entire article

- Common Tags – div, article
- Common classes – story, main, content, article

Header - Contains the Article heading

- Common Tags
  Highly used- H1, H2
  Rarely used - div, p
- Common Classes - head, title

Time - Contains the time when the article was published

- Common Tags – span, p, div, time
- Common classes – date, time, datetime

Content - Contains the main story of the article
(typically contains text, image, video, ads)

- Common Tags – div, p
- Common classes – main-story, content, article-body

Figure A – Structure of a common news article

As the volume, and the variety in the content to be scraped is humongous, deciding on the technical programming stack for development is critical. The next section briefs about the technical specifications and the libraries used to bolster them.

IV. Technical Specification

1. Programming Language - Python
2. Python Libraries
   a) PhantomJS - Used to make requests to websites and receive the content by making python environment as a Phantom browser. This helps in avoiding the bot-blocks in few websites
   b) BeautifulSoup - Used to parse the HTML content received after getting the response

c) Pymongo- Used for connecting to MongoDB to perform CRUD[1] operations
d) gensim, rake-nltk – Natural language processing libraries to identify keywords, summary, polarity and subjectivity

3. Database - MongoDB
   To enable 24*7 data availability, the database is deployed in MLAB[2] which provides 500MB of free storage space. More information on credentials to access the database is provided in the Appendix A.

Since scraping is a time-consuming process, some database operations happen during the same time frame. Hence, the database schema is designed to maintain isolation during the database transactions. Brief description of the database schema design is discussed in the next section.

V. Database Schema

The MongoDB database "worldnews" contains 4 collections.

1. Homepages
   The static collection which holds the records of regions and the homepages. The reason for segregating the region is to establish scalability, for adding new regions and websites in the later stages of the application

Field Name	Data Type	Description
region	String	Name of the region
homepages	Array	Array of homepage URLs for the news sites

2. Scraped URL Log
   The scraped URL log collection holds the log information of keyword searched, the time of search, and the news article's URLs fetched after google search for that keyword. The log_time in this document is used for checking the relevance of the scraped data based on user input

Field Name	Data Type	Description
keyword	String	Keyword used to search for the article
log_time	ISODate	The latest fetched time for the keyword
urls	Array of Documents/Dictionaries	Homepage URL and the searched URLs from the home URL in an array

3. Articles
   The scraped articles are stored in this collection along with the time scraped. The header, content and the time when the article was published are scraped and stored. Depending on the nature of the article, few scrapings resulted in empty set or irrelevant content. The filtering of the articles is done while deriving insights

[1] CRUD – Create, Read, Update, Delete Operations
[2] MLAB – http://mlab.com

Field Name	Data Type	Description
url	String	Link pointing to the news article
time	String	Time when the article was published
content	String	Content of the news article
parent_keyword	String	The keyword which was used to search this article
header	String	Title of the article
scraped_time	ISODate	The latest scraped time for this article

4. Insights
   This collection is used to store the insights derived from the processed articles

Field Name	Data Type	Description
url	String	Briefly discussed in further sections
processed_time	ISODate	
summary	String	
polarity	Number	
subjectivity	Number	
keywords	Array of Strings	
parent_keyword	String	
region	String	

The database is layered in such a way to enable,

a. Storing semi-structured data like sub documents and arrays
b. Access of individual collections independently without dependencies

VI. Crawling Process

The crawling process takes 2 inputs and crawls through the set of articles obtained after the searching process. Once crawling is done, the data is filtered, and insights are discovered. On an average, crawling for each keyword (search term) takes 20-30 minutes which explains the complexity of the process.

1. Inputs

a. Keyword – String input for which the news articles must be scraped.
   Ex. “Singapore Trump Kim Summit”
b. Time Delta – String input representing the acceptance level for the relevance of data. It is represented as “<<NUMBER>><<UNIT>>”.
   Ex. 1D represents 1 day, and 2W represents 2 weeks. Valid inputs are D – Days, M – Months, W – Weeks, Y – Years

Based on the time delta input, a check point date is found, and if there is any scraped data available after the check point date, then the results are fetched. If the keyword is not present within the check point date in the database, a google search is made to find articles from the news websites.

To make the search specific to website and keyword, the site search syntax from google search is used.

Site Search Syntax - site:<<URL>> <<KEYWORD>>

Example – site:http://news.com.au FIFA world cup

Every search typically results in 8-10 articles and in turn for every keyword the number of articles scraped will be around 160-200 based on the nature of the keyword. Each of these articles are scraped and stored in the database.

2. Crawling Workflow

An overview of the entire crawling workflow is depicted below,

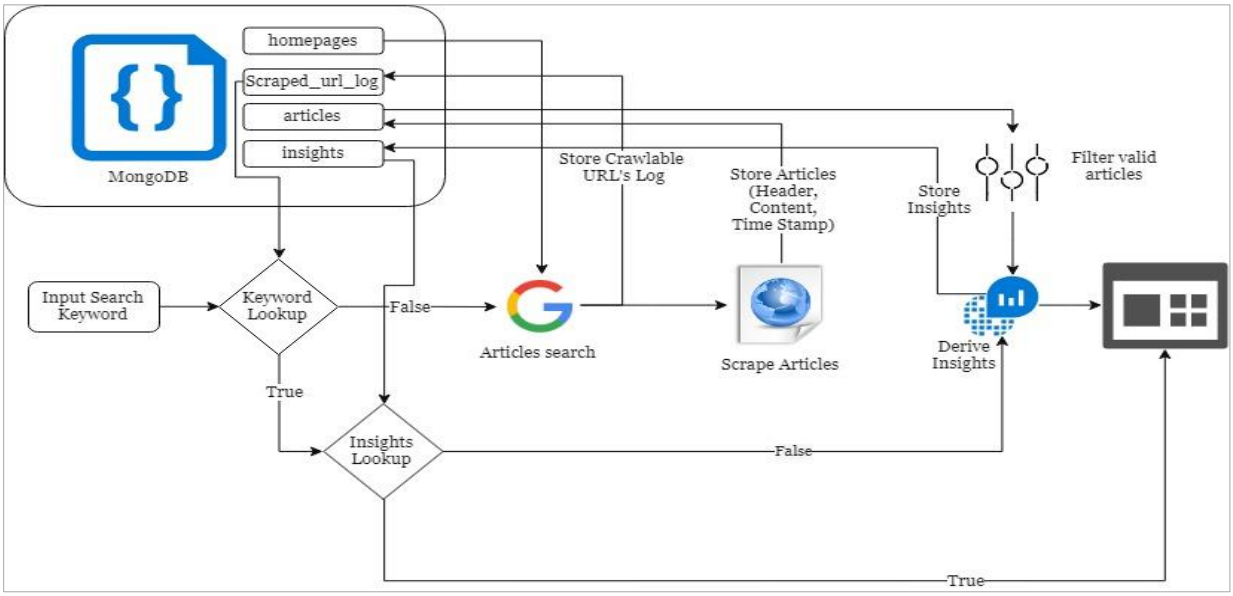


Figure B – Crawling Workflow

3. Key challenges

a. Variety of tags used in 20 different websites
   Generalized algorithm including multiple nested conditions
b. Google search blocked after certain number of repeated requests and Firewall block for repeated network usage in short span
   Introducing time delay between every request
c. Relevant content from news article
   Scraped article is checked for amount of text in the article. i.e. more than 200 words or 10 sentences. Few irrelevant information-like ads, social media sharing options, reference to related articles, copyright information, author information (irrelevant for current scenario) are safely excluded by using the classes of tag to which those content belongs
d. Article Published Date
   Representation of date varies from website to website. Below are some of the different representations of published time. Since the time conversion libraries are marginally out of scope for this project, the time is directly stored as a string in the database. Some example time representations are:
   i. May 27, 20186:11am,
   ii. Monday, 20 August 2018
   iii. Tuesday, August 21, 2018

The entire process is described in the crawling algorithm in the next section.

VII. Crawling Algorithm

Figure C provides the overall algorithm for the scraping process,

1. START
2. INPUTS –keyword, timedelta
3. IF keyword in scraped_url_log and currentdate-scraped_time<timedelta
    a. If count(insights)>0
        i. Fetch insights
        ii. Load output.json with insights
    b. Else
        i. Fetch articles scraped with required relevance (currentdate-timedelta)
        ii. Derive insights (summary, polarity, subjectivity, keywords)
        iii. Save insights along with the processed time
        iv. Load output.json with insights
4. Else
    a. Fetch all homepage urls
    b. For every homepage url
        i. Search in google using “site:url keyword”
        ii. Scrape all the results from google to find article URLs
        iii. Save the scraped urls in scraped_url_log
        iv. For every scraped URL
            1. Fetch the content
            2. Find the header
            3. Find the time
            4. Find the article container/content
            5. Save the article along with scraped time, url and the keyword searched
        v. Continue step 3b
5. STOP

Figure C - Crawling Algorithm

The following four insights are determined for each article,

1. Summary - Brief gist of the content
2. Keywords – Content specific keywords
3. Polarity - The expressed opinion in a document. A sentence or an entity feature/aspect is positive, negative, or neutral on a scale of 0 to 1 where 1 is positive
4. Subjectivity - The subjective nature of the document on a scale of 0 to 1

The downloaded insights in the form of JSON are loaded in tableau for data visualization.

VIII. Data Insights

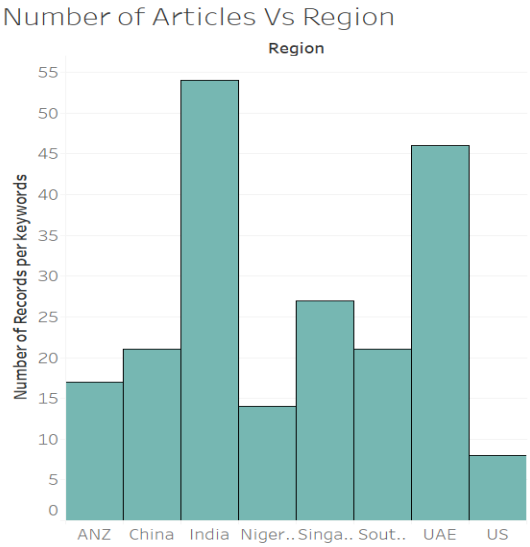
The JSON file is loaded in tableau environment and below insights are visualized.

1. Word Cloud – Keywords

Provides information on overall or region wise most used words in the news articles for a specific search topic

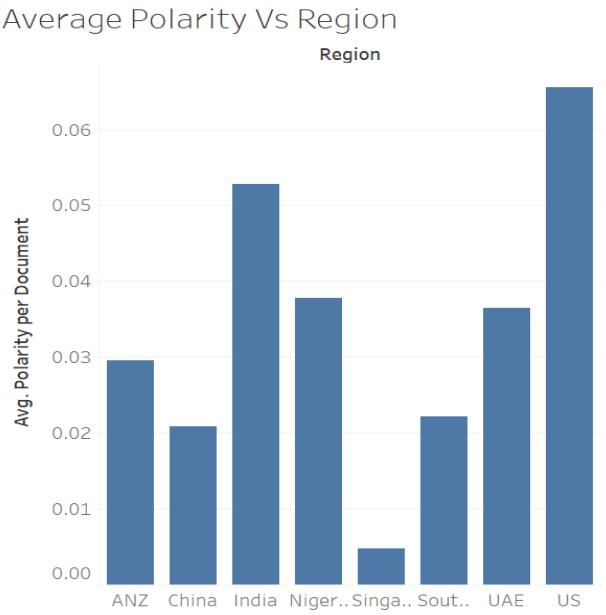
2. Region-wise number of articles

Provides the cumulative number of valid articles (after filtering) for each region. The below bar chart represents the number of articles fetched for Kerala (Indian State) flood. It is observed that after India, UAE has more articles published for the same. UAE government has also announced 700Crore INR as a relief fund for the flood affected areas. Understandably more people from the state of Kerala are now permanent residents in UAE



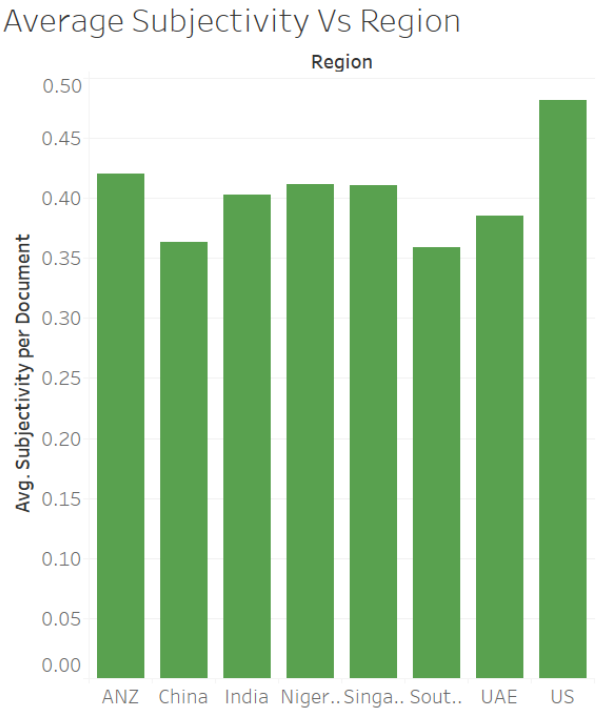
3. **Region-wise polarity of the news**

This plot explains how the news is delivered to people. A marginal difference can be observed for different news across different regions depending on the keyword



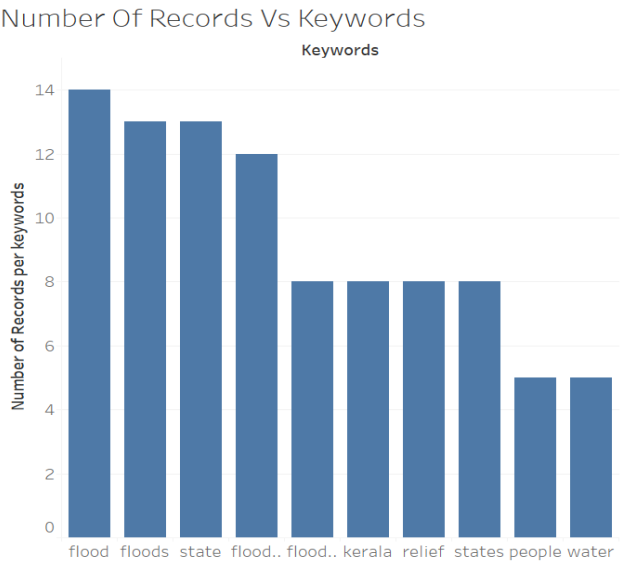
4. **Region-wise subjectivity of the news**

This plot explains how subjective the news articles are. For multiple keywords, almost all the regions fall under the same band for subjectivity



5. **Most used Keywords**

This plot explains the most used keywords with the number of articles possessing the keyword.



Refer to the attached "Kerala Floods.twb" file for the tableau workbook. All the above insights have a fewer regions since few regions are manually excluded because of irrelevant news. Refer to "Cambridge Analytica.twb" for another set of insights related to "Cambridge Analytica"

IX. Discussion and Future work

1. The articles scraped can be preprocessed further before storing to enhance the insight discovery
2. Along with the scraped time, the article's published and updated time also can be considered for filtering which typically depends on the nature of the keyword searched for. For example, news like "Kerala floods" have very recent articles published in the recent times whereas news like "Singapore Trump Kim summit" will have articles which are old
3. Despite google search being a good start point to identify articles, direct search on the specific news websites will provide more relevant news which again needs the presence of search option in the news sites
4. Publicly available API's for fetching news articles can be used to get the precise articles
5. Identifying and filtering fake news is a peculiar research area which can be included in the future

X. References

1. http://www.bandt.com.au/media/2018-year-online-media-consumption-overtake-traditional-media
2. http://assets.pewresearch.org/wp-content/uploads/sites/13/2018/07/11183646/State-of-the-News-Media_2017-Archive.pdf
3. https://reutersinstitute.politics.ox.ac.uk/sites/default/files/Digital%20News%20Report%202017%20web_0.pdf
4. https://www.mongodb.com/blog/post/6-rules-of-thumb-for-mongodb-schema-design-part-1
5. http://mlab.com
6. https://docs.mongodb.com/manual/tutorial/getting-started/

XI. APPENDIX

A. Data Base Access

A "read only" public account is created for researchers reading this report to view the data.
Connection host – ds119442.mlab.com
Port – 19442
Database – worldnews
User name – public (case sensitive)
Password – visitor4worldnews (case sensitive)

B. Description of Modules & Functions (attached application.zip)

Module	Function	Description
main.py	start_process(keyword,timedelta)	Starting point of the application which orchestrates the entire work flow of the application
search_phantom.py	keyword_search(keyword)	Receives the keyword and preforms google search with homepages and returns the article URLs
news_scraper.py	load_data()	Loads the articles URL in a global variable so that it can be accessed by other functions
	parse_articles(all_urls,keyword)	Loops through all article URLs and obtains the parsed content
	parse_content(url)	Obtains the article from URL, and passes the content to find header, content and time
	find_header(soup_content)	Finds the header of the news article
	find_time(soup_content)	Finds the time of the article published
	find_content_text(header,soup_content)	Finds the article content
db_service.py	get_regions_with_news_sites()	Returns the regions along with home page of news sites
	connect_to_db()	establishes connection to mlab MongoDB database
	close_connection()	Closes the connection to mlab MongoDB database
	store_region_and_urls()	Stores the regions along with home page of news sites
	check_if_keyword_present(keyword,timedelta=None)	Returns True/False based on keyword availability in scrape_log
	get_insights(keyword,timedelta=None)	Returns the insights as JSON array
	get_articles(keyword,timedelta=None)	Returns the articles as JSON array
	save_urls(all_urls,keyword)	Save the urls in the scrape_log
	save_insights_to_db(insight)	Saves the insight to db
	Save_articles_to_db(article)	Saves the article to db
analyser.py	analyse(articles)	Loops through all the articles to find insights
	filter_content(articles)	Filters the articles based on content size
	keyword_extraction(text)	Returns keyword from the given text

	summarize_and_extract_keywords(text)	Returns summary and keywords for a given text
	find_polarity_and_subjectivity(article_content)	Returns polarity and subjectivity score for the given text
Utils.py	validateURL(url)	Returns True if the given URL is valid
	get_check_date(timedelta)	Returns a date, which is the check point date based on time delta passed
	exclusions()	Returns the list of classes irrelevant to the content
	get_region_for_url(url)	Returns the region for the provided url
	download_insights(insights)	saves insights as json file

C. Execution guide –

Run the main.py file from command prompt by passing the inputs as command line arguments.

Syntax – python main.py “Keyword” “TimeDelta”

Example – python main.py “US foreign policy” “2D”