Programming for Big Data Analytics

Project Report

News Dashboard

Aakaash Jois - aj2309
Bhavana Ramakrishna - br1525
Gahan Jagadeesh - gj717
Mohammed Affan Vuppukar - mav487

Abstract

News Dashboard is a curator of news articles from various sources on the internet to provide the user with streamlined and compact news articles with deep insights. In this era of time constraints, most individuals have minimal time to invest in reading huge news and media articles. News Dashboard focusing on solving this problem. This project focuses on creating an architecture to curate, simplify and provide the news articles to the user when queried.

Contents

1 Introduction

News is the cornerstone of our civilization. Not only does it give us information on what is happening around the world, but it also shapes our perception and opinions of the events around us. However, in the digital age, we are flooded with more articles in a day than we can read in a lifetime! Furthermore, a significant number of these articles are just click bait which serves no purpose to the user. This makes it hard for people to sift through articles which they find relevant and useful. It is also a waste of time to go through the entire article considering the fact that articles have a lot of fillers which are not important. So, the fillers need to be filtered to extract the summary of the article.

In order to tackle these problems, we created large-scale article summarizer. This system gives us summaries of articles and is able to scale to handle a large number of articles by distributing the load among different nodes. By giving people the summary of articles we believe we can make people choose which articles to read more wisely and help them avoid click bait articles. People would no longer have to peruse the entire article to get the gist of what it is trying to convey, but can just read the summary based on the categories. Our model only returns the important parts of the article which is just about enough to summarize the entire article. The model also displays a list of keywords which are frequently used in articles in certain categories such as from a single state or a single country which will enable us to deduce a pattern with respect to the huge amount of articles available to us.

2 Methodology

2.1 Architecture

Docker is used to run the containers where we create a Python, Spark and a Mongo container. The architecture of the model begins with the Python containers in which Dispy distributes the tasks to various nodes. The data extracted from the python container is then passed to MongoDB where it is stored as a NoSQL Database. Next, The data is sent to Apache Spark and then Data cleaning and analysis is performed by Spark. Spark makes every process scalable. Then the analyzed data is used in Tableau to create visualizations.

2.2 Data Scraping

In this project, the Indian news sites are taken as the data source. We are primarily using The Hindu and Times of India, the two biggest news sites in India. We have acquired the RSS feeds based on category and also based on the state for these sites and have aggregated them into a CSV. We can easily add more RSS feed if needed to scale.

We have used a framework called Dispy [9] which is used to distribute tasks to a cluster of worker nodes using a client node. We use this framework to distribute RSS feeds to a cluster of worker nodes. These worker nodes use the

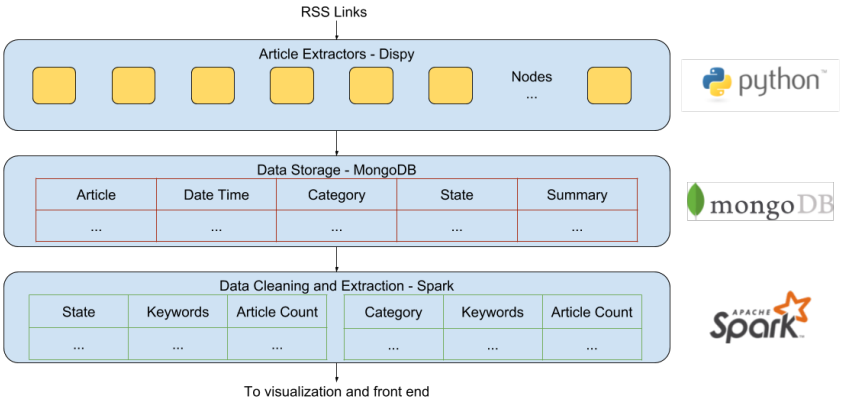

Figure 1: News Dashboard Architecture

Requests and BeautifulSoup [10] frameworks to parse through the RSS feeds and get the articles. This architecture is easily scalable as we can add more worker nodes in the cluster to increase throughput.

2.3 Summary Extraction

Summary extraction takes place in the worker nodes directly after retrieving the articles. We use a framework called Sumy [8] to achieve summarization. We use LSA [6] summarizer to attain a summary of the article. Stop words are passed to the summarizer to perform effective summaries. In addition, we enable the summarizer to perform stemming in order to further optimize our summaries.

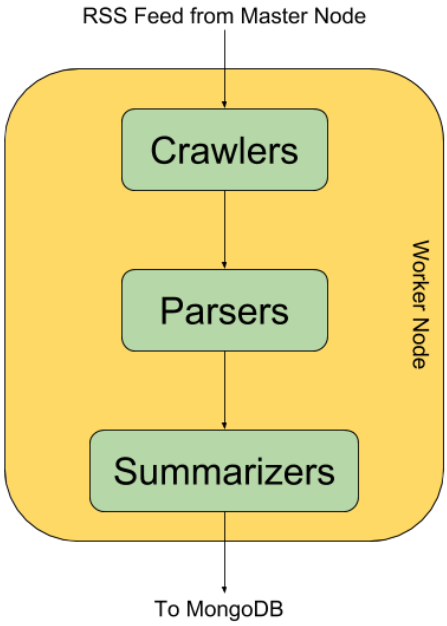

Figure 2: Node Architecture

Article

... Banking on an eclectic mix of tourist destinations across Kurnool district, officials of the Tourism Department are planning not only to improve the existing facilities but also to introduce new attractions to exploit the untapped potential of the region. Taking a step forward in this direction, the officials have reviewed the status of tourist amenities and embarked upon improving them to suit the present demand, besides focussing on developing new locations. First among the series of initiatives is developing Allagadda, one of the major stone carving centres in the State, into a "Sculpture hub", as envisaged by State Tourism Minister Bhuma Akhila Priya. The officials plan to establish working areas for the traditional sculptors, selling counters (including examining the possibility for export) and a state-of-the-art \u2018Cultural Auditorium\u2019, to highlight the importance of the region. The developments are also aimed at improving the lifestyle and prospects of the traditional sculptors. The Allagadda stone carvings have been bestowed with a Geographical Indication (GI) tag. Another major emphasis is on the aspect of "Adventure Tourism". While the Orvakal Rock Gardens boasts hosting a series of activities such as rock climbing, zipline, bungee trampoline, rappelling etc., the officials have identified an unexplored underground cave system \u2014 Valmiki Caves in Boyavandlapalli village of Peapully mandal \u2014 and have enlisted the help of Italian speleologists to examine the feasibility of opening it to the public. The cave system\u2019s narrow entrance, which can only be accessed after a strenuous trek, treacherous drops of varying levels, rock formations, pristine water bodies and much more, would make it an ideal adventure spot, if everything goes according to the plan ...

Summary

u'Banking on an eclectic mix of tourist destinations across Kurnool district, officials of the Tourism Department are planning not only to improve the existing facilities but also to introduce new attractions to exploit the untapped potential of the region.The officials plan to establish working areas for the traditional sculptors, selling counters (including examining the possibility for export) and a state-of-the-art \u2018Cultural Auditorium\u2019, to highlight the importance of the region.The cave system\u2019s narrow entrance, which can only be accessed after a strenuous trek, treacherous drops of varying levels, rock formations, pristine water bodies and much more, would make it an ideal adventure spot, if everything goes according to the plan.'

Figure 3: Article and Summary

The summarizer gives us the option to choose the number of sentences to summarize in which we can use to give users concise summaries. This does take some time to compute but since we have distributed the work among a cluster of nodes, we have effectively increased the throughput. The result from the summarizer is shown in Fig.3.

2.4 Data Storing

For Data Storage, we use MongoDB [3]. We chose MongoDB as it is a NoSQL database which can be distributed among several nodes and is easily scalable. We use the PyMongo [4] framework to connect each worker to the MongoDB database. Each worker node is connected to the same MongoDB database and after the summaries are performed we store the summary, article, state, category and the datetime for the article. This information is appended to the end of the MongoDB table and as each transaction is atomic, we are able to concurrently perform transactions from each worker node. By allowing multiple worker nodes to simultaneously store in the database, we are able to scale up the number of worker nodes and can effectively increase throughput.

2.5 Data Processing

The data stored in MongoDB is then passed through the Apache Spark container created from Docker. MongoDB is connected to spark using MongoDB connector for Spark. The data is then pulled from Mongo and then stored in a dataframe to be used in spark. The Spark [2] DataFrame then holds the Articles. Then, we remove the stop words to eliminate the filler words such as the, and, or. We then use lemmatization to determine the lemma of the words so we do not get verb duplicates. After the articles are simplified, We then we perform word count on the articles to get the important keywords categorized by State and Categories.

3 Visualization

Finally, we visualize the processed data by encoding it as visual objects with respect to various parameters to get an interactive overview of what the model achieves. The data is compared with each other to get a visual representation of various trends and changes in the data. We use Tableau [5] to handle the visualization by using the input as CSV files obtained by exporting the Spark Dataframes. The top few keywords encountered in the articles (Fig.4), the number of articles per categories (Fig.5 and state (Fig.7) are a few of the parameters considered while visually representing the data. The data over a period of time is also compared to analyze the change in trends of the articles over time (Fig.6).

Figure 4: Keyword Count

Figure 5: Number of articles per category

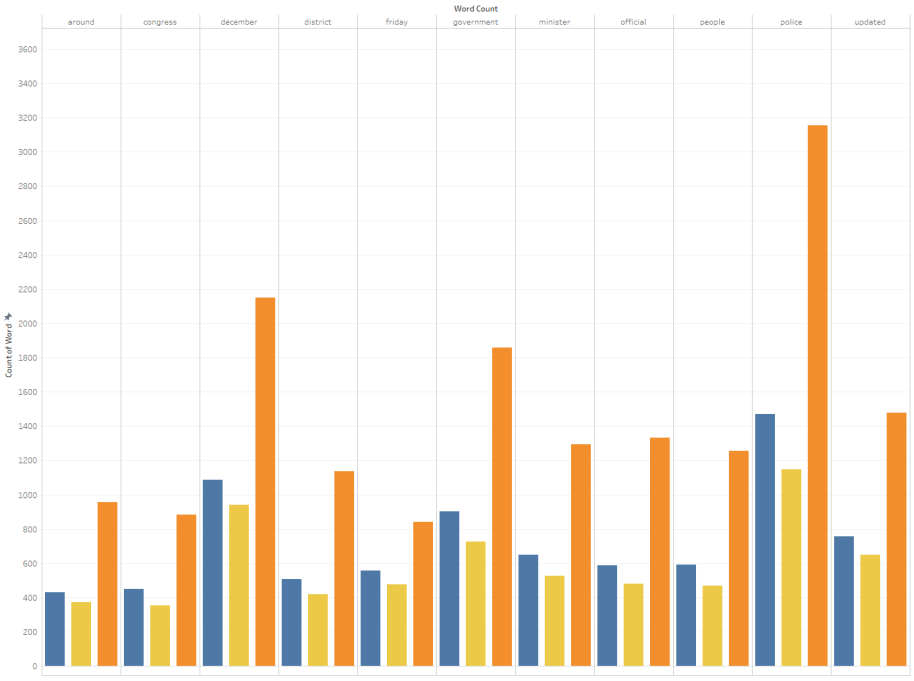

Figure 6: Keyword trend over 3 days

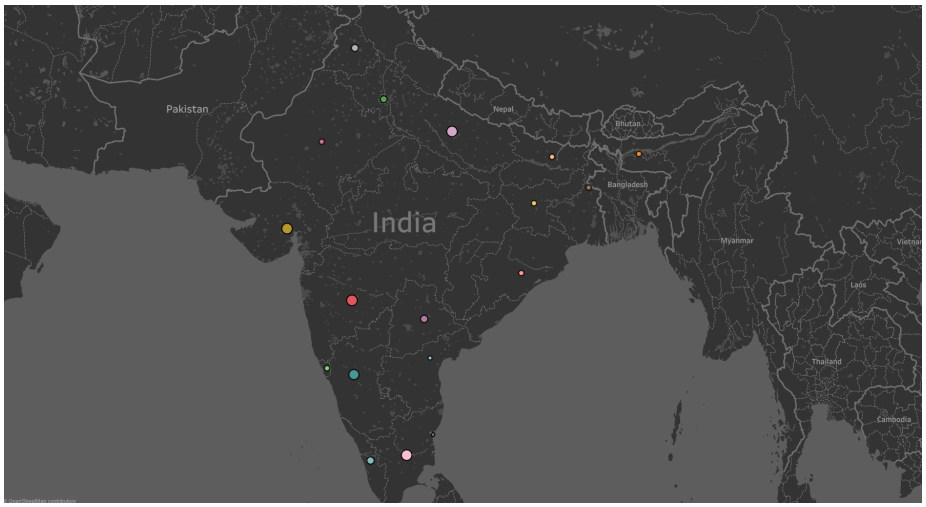

Figure 7: Number of articles per state

4 Obstacles

There were a lot of detours we had to take while coming up with this model. One of the primary challenges we faced was making the model scalable. There were a lot of different approaches before we ended up with using Dispy, Mongo, and Spark to make the model scalable. Also, running all the containers in Docker and then linking them was a challenge we faced which was overcome by creating a docker network and storing the containers in them. To overcome the hardware requirements for this project, We tried to use NYUs High-performance computing (HPC) division to run our model but the Mongo was only able to run on NYU Prince and Spark was only able to run on NYU Dumbo. They were running on two different clusters in the HPC environment and there was no way to link the two to complete our model.

5 Takeaways

One of the primary and most important things that were learned from this project was to make a model scalable and run it in a distributed environment. This enables the model to be run in various different systems in a distributed way and still arrive at the same result. This project also required us to be able to link several different containers and tools to be able to transfer the data around a cluster and also perform analysis on the data which was passed through multiple tools. A key takeaway from this project was, we learned how to create a scalable architecture for handling big data.

6 Future Work

There are a number of directions we would like to pursue in the future. Firstly, we plan to add news articles from multiple outlets across the world, covering various categories. We would like to implement the entire model on AWS platform using various microservices - AWS Kinesis [1] to perform real-time data streaming as and when the article is generated, and trigger data processing and visualization once in every hour, thus reflecting real-time data visualization. We also plan to include more models to visualize global data to perform analysis on articles over a certain period of time. Lastly, since the articles will be stored within the environment, we would like to implement Elastic Search [7] on the news data to search based on different categories or even query using keywords.

7 Conclusion

So, the article summarizer is run successfully in a distributed framework across multiple nodes thus increasing the throughput. The article summaries were stored in a NoSQL Database (MongoDB) along with the categories and states for further use. The articles stored in MongoDB are then filtered to simplify

them to only process the important information. This is achieved by removing stopwords and lemmatizing as described in the previous sections. Finally, the cleaned data is used to extract keywords and perform data analysis. The output is then pushed to Tableau to perform visualization.

8 Code

The code used for this project is attached as a zip file with this report. It is also available on GitHub at https://github.com/aakaashjois/News-Dashboard.

References

[1] Amazon kinesis.

[2] Apache spark - unified analytics engine for big data.

[3] Open source document database.

[4] Pymongo 3.7.2 documentation.

[5] Tableau: Business intelligence and analytics software.

[6] S. T. Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[7] C. Gormley and Z. Tong. *Elasticsearch: The Definitive Guide: A Distributed Real-Time Search and Analytics Engine.* " O'Reilly Media, Inc.", 2015.

[8] miso belica. miso-belica/sumy, Apr 2018.

[9] G. p. Pemmasani. dispy, Dec 2018.

[10] L. Richardson. Beautiful soup.