Wikipeaks: Events Detector

Osvaldo Bulos*, Shixin Li* and Zewei Liu*, *Center for Data Science, NYU*

Abstract—This project focused on building a system that helps to identify daily global and local events by using the pageviews of Wikipedia articles in different languages. The team designed and implemented the architecture to store, process and analyze the huge amount of information provided by Wikipedia. This architecture runs over the Amazon Web Services. The team also developed a function to rank daily peaks in the pageviews taking a time-window of 15 days. Finally, a web system was implemented to visualize the correlation between an article's peak and its related events for a specific day.

Index Terms—Peak Detection, Wikipedia Pageviews, Web Service, Detection of Global and Local Events, Data Visualization

1 Introduction

Wikipedia is the largest and most popular general reference work on the Internet. It compresses millions of articles in different languages and has already become one of the most important and essential website to nearly everyone in the world who have access to the Internet.

Although Wikipedia has a reputation of an academic and reference resource, the constant aggregation of new articles involving any possible topic of human life and the millions of views it receives every day are factors that can be used as an indicator of major global and local events.

An important point to consider is that Wikipedia present several advantages over traditional or more conservative news agencies. Wikipedia, being an open tool that can be accessed and edited by any person at any time, allows the articles to be less biased and more objective towards specifics events.

Based on all these advantages, it is sufficient and reliable to acquire the world trend from the change of the Wikipedia pageviews.

2 Related Work

Many tools have already been developed to analyze or visualize the Wikipedia pages. For example, Wikipedia has a tool to analyze the statistics of the page views of one or several articles. [1] Wiki features, data, and meta-data can be accessed via MediaWiki action API [2] .

Also, there is on-going research in the peak detection field. For example, Azzini et al (2004) has done some work for peak detection on the bioinformatics problem using time series microarray data[3]. Harmer et al (2008)) [4] has applied a peak-trough detection algorithms to image processing. Girish Palshikar(2009) [5] has written a review to formalize the notion of a peak in a time-series environment and classified several algorithms for peak detection.

For this project, the team is trying to combine the analysis and statistics of the Wikipedia pageviews and apply an original peak detection algorithm. However, the amount of the data from Wikipedia ranges in the millions making it a difficult task to process and analyze. To handle huge amounts of data, Jeffrey Dean et al(2004)[6] has proposed a MapReduce framework. Besides that, Amazon Web Services also offer cloud computing resource to address big data problems. The team will evaluate the NoSQL databases approach, which has become popular in recent years for being more stable and faster compared with traditional databases. And finally, HTML5 and Javascript provides a faster and more beautiful way to visualize data.

3 Project's Description

The goal of the project is to build a system to identify global events by using the pageviews of Wikipedia articles in different languages. The baseline model will use the English articles, since it has the largest corpus among all the languages. The Top Events/Articles will be identified from this subset and then compared with the same article in other languages. It was decided just to compare the English articles with the following most spoken languages in the world: Chinese and Spanish. French was added too.

The project will have the following workflow:

1) For a specific day, the views of all the English articles visited that day will be obtained.
2) A scoring function will be applied to each of the articles previously identified as a peak in the selected day.
3) The articles will be ranked based on their scores.
4) A Top N articles will be selected.
5) Each of the articles in the Top N will be able to be compared with their matching articles in Spanish, Chinese and French (in case these articles exist).

4 Implementation

4.1 Data and Preprocessing

Wikipedia releases every hour the Pageviews of their articles visited in that window of time. The data sets can be downloaded at [9]. There is information about the views starting from May 2015 until the present day.

Each file has the following structure:

* *Team members contributed equally to this project.*

Sample Data From October 1st, 2016
From 12:00 to 1:00 am.

```
en.m Gatumba 1 0
en.m Gatun_Lake 2 0
en.m Gatundu 1 0
en.m Gatwick_Airport 11 0
en.m Gatwick_Airport_railway_station 3 0
en.m Gatwick_Express 3 0
en.m Gau_(territory) 1 0
en.m Gau_Saxony 1 0
```

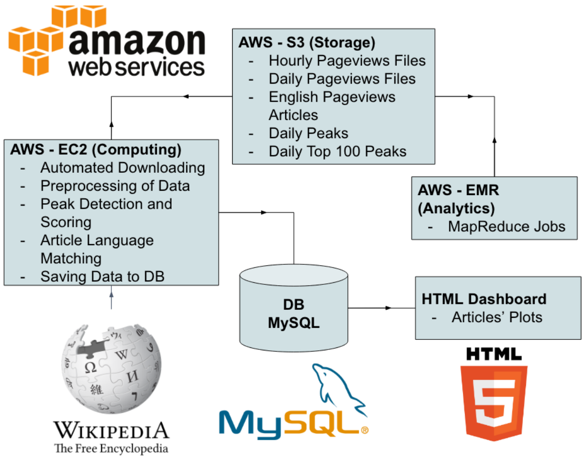
TABLE 1: Data Format

Section	Meaning
Language Code	Following the ISO-639-1 Standard
Wikipedia Project	wikibooks: ".b" wiktionary: ".d" wikimedia: ".m" wikipedia mobile: ".mw" wikinews: ".n" wikiquote: ".q" wikisource: ".s" wikiversity: ".v" mediawiki: ".w"
Page Article	Internal name used to access the Wikipedia article.
Hourly Views	Calculate hourly pageviews.
Arbitrary 0	No information about this value in the documentation. height

4.2 Project's Architecture

Once the datasets were analyzed, different challenges were identified. The principal and most obvious was the size of the dataset. Each hourly file contains about 4 million rounds, which translate to 4GB per day. This amount of information required special attention to find the right platform for storage and computing capabilities. A decision was made to limit the data for the project and only use the views from 2016: from January 1st to November 1st. However, this still represented a challenge: 24 files per day × 305 days = 7320 files = 1220 GB = 1.22TB The second challenge was to identify a way to automate the downloading of 7320 files and preprocess them, given that the files contain data by hour, while the time unit for the project was set to days. The goal was to minimize the number of human-triggered actions.

After analyzing the challenges and constraints and evaluating different options, the Amazon Web Services (AWS) were selected to develop the project. The main factors where: AWS provides a reliable and robust solution for Storage, Computing, Analytics and Deploying Services. AWS modules are scalable. AWS provides APIs and commands to interconnect modules and services.



Using the above architecture, the data flowed in this way:

1) Automated Downloading of Pageview files from Wikipedia.
2) Pageview files stored on AWS S3 Buckets.
3) Preprocessing / Filtering of Pageview Files
4) MapReduce jobs to generate Pageviews per Day
5) Save the output data on a DB
6) Peak Identification in Pageviews
7) Data displaying through an HTML Dashboard

4.3 Pages Matching Method

Another challenge that the team faced, was to match the same articles in different languages. This is not a simple task due that the name of an article in English is not mapped directly to other languages.

For example:
https://en.wikipedia.org/wiki/Olympic_Games
Does not match directly to the spanish wikipedia like:
https://es.wikipedia.org/wiki/Olympic_Games
But instead to:
https://es.wikipedia.org/wiki/Juegos_Olímpicos

We can see that the names of the articles are in the language of that version of Wikipedia.

The solution for this problem was the following:

1) Select an English article.
2) Crawl the English Wikipedia page https://en.wikipedia.org/wiki/(English_article_name).
3) Find the Languages Container and use a regular expression to find the corresponding name in Spanish, Chinese and French.

However, the team did not take in consideration the different encodings for accents and other symbols from those languages, so when obtaining the article's name in another language and tried to pair it with the articles in our datasets, there was still a mismatch sometimes. Unfortunately, the time constraints limited us to solve this issue.

4.4 Scoring Function

$$score = \lambda_1 \times popularity + \lambda_2 \times change\_rate \tag{1}$$

$$popularity = \frac{pageviews}{average\_pageviews} \tag{2}$$

$$change\_rate = \frac{pageviews - 14\_days\_average\_pageviews}{14\_days\_average\_pageviews} \tag{3}$$

$$\lambda_1 + \lambda_2 = 1 \tag{4}$$

So far, the scoring function has two parameters: popularity and change_rate, as these two parameters can be two key factors that determine how important a peak is. For each day, since there are thousands of pages that reach their peaks, a scoring function should be involved in to select pages that are useful and meaningful, thus make it easy to locate some important events that have happened or are happening in some regions of the world based on these pages.

Popularity:

1) pageviews is the number of the views.
2) average_pageviews is the average number of the views of the popular pages (This means that the average_pageviews is not the average views of all pages; instead, it is only the average views of the pages which views are larger than 1000).

By having $popularity = \frac{pageviews}{average\_pageviews}$, this parameter can be implemented to only keep pages that are at least popular enough, so that can be seen as events related pages for a specific day.

Change_rate: the change_rate is to measure the percentage change of the views in a selected day by comparing with other 14 days within a 15-days window . This parameter is applied to find out the pages, which their number of views increase largely in the selected day.

λ_1, λ_2: λ_1 and λ_2 are two weight coefficients used to combine popularity and change_rate.

4.5 Databases

Once the Top 100 Peaks files were obtained, two Database options were evaluated to feed the HTML dashboard. The first option was DynamoDB, a NoSQL database provided by Amazon and MySQL as second option. Both services are provided in the AWS ecosystem. DynamoDB, being a NoSQL database presented various advantages:

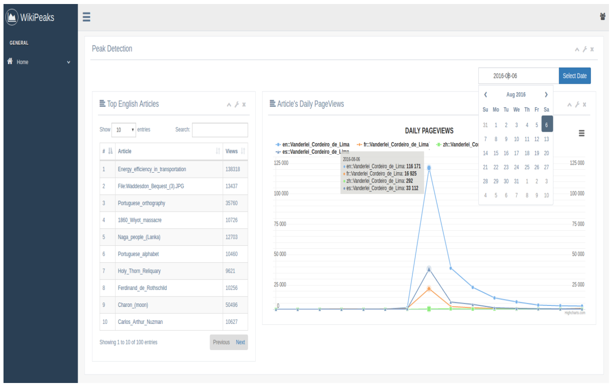
1) Build to handle the requirements of Big Data applications.
2) No need to define a Schema.
3) Can handle unstructured data.
4) The scaling of the DB is cheaper.

However, after several test, the team noticed that importing the data into DynamoDB was a slowly process with different limitations: The database has a number of Write and Read operations per minute (Exceeding the number of operations will incur in costs by AWS), the data needs to have a special format and the batch import is limited to 25 elements per request. Along these issues, there are not many options available for connectors between Python and DynamoDB (just BOTO, provided by AWS). MySQL was evaluated as well, and the team decided to use this database instead for the following reasons:

1) Even that the Wikipedia daily articles contains millions of data points. After applying the scoring function and selecting the Top 100 articles, the data was reduced to only 100 * 306 = 30,600 data points. Quantity that can be easily managed by a relational database.
2) The data obtained has a common structure: Date, English Article, 15 days Pageviews, 15 days Pageviews for the French article, 15 days Pageviews for the Chinese article and 15 days Pageviews for the Spanish article.
3) Since MySQL is quite a popular database, there are many options available for web connections as long as technical support.

4.6 Web System

A Web System was implemented to present the Top 100 articles per day in a friendly way to the users. This system follows an Model View Controller (MVC) architecture to allow the connection to the database, query and prepare the data and display it through an HTML website. The Django framework was used to achieve the above. Django allows an easy integration between Python, database connections and HTML. For the visualization, HTML5 and Javascript was used to display an interactive table to show the Top 100 articles and a line plot for the pageviews of the 4 different languages. A user can select a date in a calendar (which is limited from January 1st to November 1st). The data will be be loaded from the database and populate the dynamic table. From here, the user, by clicking an article, can see its pageviews for the 15 days period window for the 4 language articles. In case that the only peak is from the English article, it can be said that a local event happened, but if the peak appears in all 4 languages, then a global event has been detected.



#	Article	Views
1	Energy_efficiency_in_transportation	138318
2	File:Waddesdon_Bequest_(3).JPG	13437
3	Portuguese_orthography	35760
4	1860_Wiyot_massacre	10726
5	Naga_people_(Lanka)	12703
6	Portuguese_alphabet	10460
7	Holy_Thorn_Reliquary	9621
8	Ferdinand_de_Rothschild	10256
9	Charon_(moon)	50496
10	Carlos_Arthur_Nuzman	10627

Showing 1 to 10 of 100 entries

2016-08-06
- en::Vanderlei_Cordeiro_de_Lima: **116 171**
- fr::Vanderlei_Cordeiro_de_Lima: **16 925**
- zh::Vanderlei_Cordeiro_de_Lima: **292**
- es::Vanderlei_Cordeiro_de_Lima: **33 112**

Fig. 1: Video is in [7]

5 Conclusion

This project successfully built a system that helps to identify daily global and local events by using the pageviews of

Wikipedia articles in different languages. On the interactive table the team created, if the only peak is from the English article, it means that a local event happened. While, if the peak appears in all 4 languages for the same article, a global event is detected.

Also, in the process of the project, the team has following findings:

1) For each important event, the peak may be reflected one day after the event has happened. (People tend to be looking for more information after the events have occurred).
2) Most of the English Articles with high scores are not necessarily popular in other languages. This is due to the higher volume of English articles compared to the other languages. The articles in English tend to be more complete generally.

For Future development, the team has the following ideas:

1) The scoring function only depends on two parameters (popularity and change rate) and can be improved by adding more factors.
2) Predict what kind of events could be a global event.
3) Compare the events detected with newspapers and social media to evaluate the results.

REFERENCES

[1]https://tools.wmflabs.org/pageviews/.

[2]https://www.mediawiki.org/wiki/API:Main_page.

[3]Azzini I., Dell'Anna R., Ciocchetta F., Demichelis F., Sboner A., Blanzieri E., Malossini A. (2004), "Simple Methods for Peak Detection in Time Series Microarray Data", Proc. CAMDA'04 (Critical Assessment of Microarray Data).

[4]Harmer K., Howells G., Sheng W., Fairhurst M., Deravi F. (2008), "A Peak-Trough Detection Algorithm Based on Momentum", Proc. IEEE Congress on Image and Signal Processing (CISP), pp. 454 – 458.

[5]Girish K. Palshikar. Simple Algorithms for Peak Detection in Time-Series. In Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence, 2009.

[6]J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In Communications of the ACM, 51 (1): 107-113, 2008.

[7]https://github.com/NYU-CDS-Capstone-Project/Whatever-OZS.