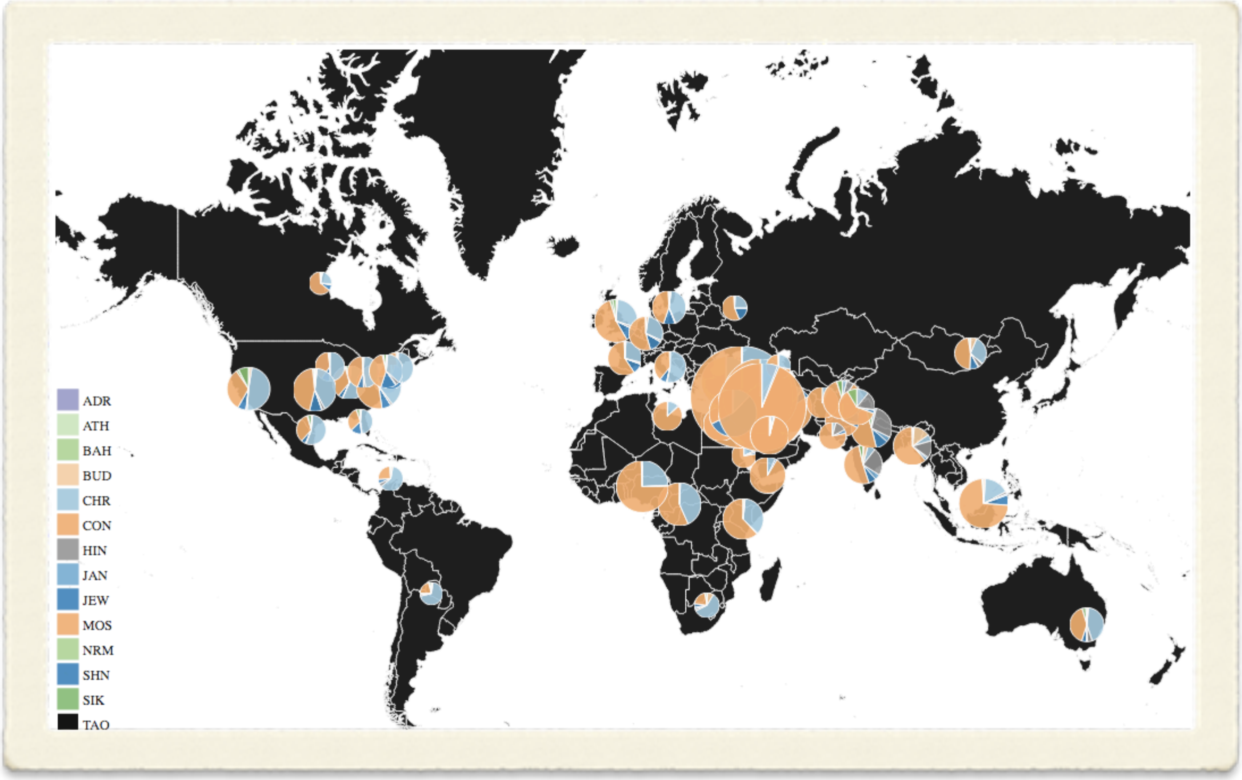
Analysing Religious Conflicts Through The Lens Of Global Media



COMP 4651 Group Project

Wong Chun Yin (20126795, JohnnyYellow)
Chandra Prashanth (20051005, prashcr)
Gupta Shivang (20165703, shivanggupta)
Tsang Hauton J (20214853, hautonjt)

Spring 2017

Table Of Contents

Introduction

Motivation:
Recent news media reports a rise in religious conflicts around the world [1]. Social scientists have also reported a rise in religious affiliation with a predicted decrease in the number of atheists by 2020 [2]. There is also a growing fear of terrorism and extremist threats like ISIS. As such, religious conflicts are of great concern in the present and the near future.

At any given point of time there are many different religious conflicts being reported around the world, however, there is also a large amount of media bias in these articles. By using a data-driven approach to analysing religious conflicts in the world, we believe we can better understand these conflicts and visualise trends.

As we will need to analyse an extremely large amount of data we hope to test the cloud analytics skills we have learnt in this course while building them further at the same time with a real use case.

Objectives:
Our primary aim is to understand the distribution of religious conflicts around the globe. By aggregating news media reports about religious conflicts from around the world, we can cluster them in order to highlight which areas have historically been more prone to religious conflicts.

We also wish to compare the news coverage of conflicts in a region vs. the actual incidence of conflicts in the area in order to see if there is a bias in the media reports.

Dataset

The Global Database of Events, Language and Tone (GDELT) Project is an open source dataset that collects broadcast, print and web news from around the world in over 100 languages and updates daily. The total size of the GDELT dataset is over 3.5 TB and it records 58 characteristics for each event that happens around the world including actors involved, location, religion and sentiment [3].

The entire GDELT dataset is available on Amazon Web Services as part of their public data sets. The data is available as a series of tab-delimited values stored in .csv files in S3 buckets. An individual day's records can be over 2 GB, as such we chose to analyse data from just the past year (May 2016 - Apr 2017) in order to avoid exhausting our free credits from AWS Educate.

Out of the 58 characteristics we chose the relevant features for our project as: time (month and year), position of event (latitude and longitude), Quad Class (used to characterise conflict events), Goldstein Scale (used to measure impact of event), number of articles reported, number of total events reported and total values for normalisation.

Cloud Setup and Overview

We chose Apache Spark over Hadoop MapReduce for our project as it is easier to implement machine learning and analytics at the same time using high level libraries like PySpark and NumPy.

We deployed a simple Spark cluster with one master and one slave on AWS EC2. We also used Amazon S3 as our storage system. To allow easy collaboration we assigned an elastic IP address to the master so that the team can ssh into the same address when we restart the cluster. The image below shows our setup in more detail:

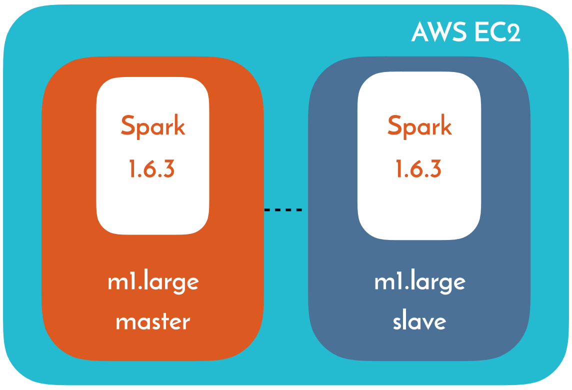


Figure 1: Server Architecture

A Python script was deployed to extract the data from S3 and store it in HDFS on our cluster. RDD operations were then performed to process this data and the output data was stored back into Amazon S3 as JSON files to facilitate visualisation.

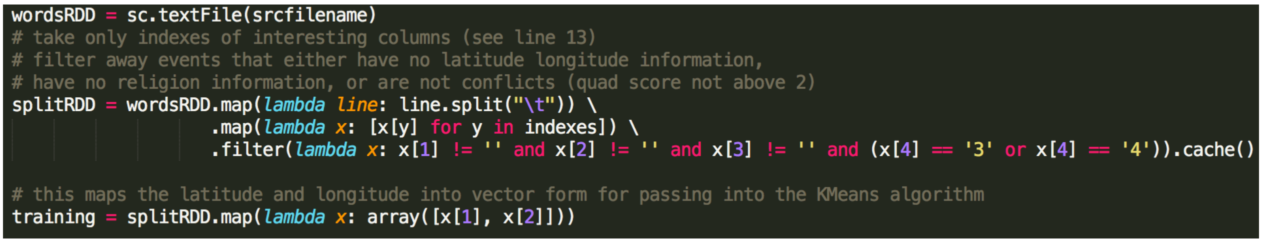
Processing Data with ML

In order to process the data and form geospatial clusters of religious conflicts, we chose to use the k-means clustering algorithm. Before the training could be done, we pre-processed the data by extracting the relevant information from the raw GDELT data files.

We first imported the raw GDELT files from S3 and then using RDD map and filter operations, selected only the relevant records as follows:

The data was then clustered using the k-means algorithm with a k value of 50 in order to highlight a significant number of clusters across the globe.

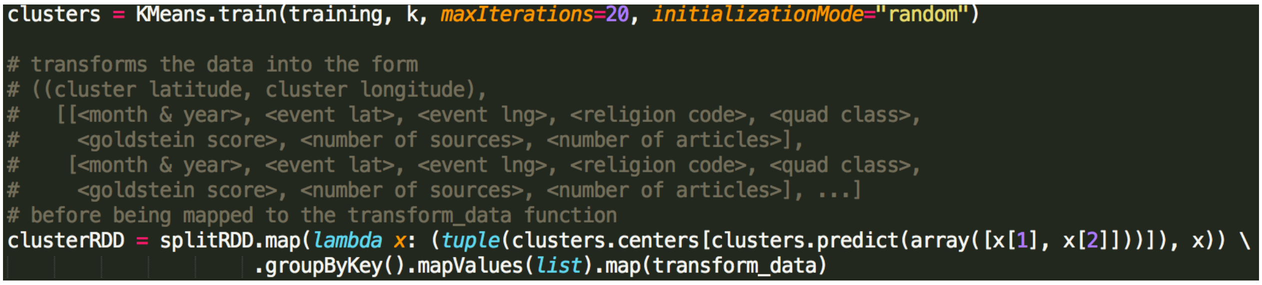
The resulting RDD was sorted and transformed, and the same process was repeated for all 12 months with the result



```
wordsRDD = sc.textFile(srcfilename)
# take only indexes of interesting columns (see line 13)
# filter away events that either have no latitude longitude information,
# have no religion information, or are not conflicts (quad score not above 2)
splitRDD = wordsRDD.map(lambda line: line.split("\t")) \
                   .map(lambda x: [x[y] for y in indexes]) \
                   .filter(lambda x: x[1] != '' and x[2] != '' and x[3] != '' and (x[4] == '3' or x[4] == '4')).cache()

# this maps the latitude and longitude into vector form for passing into the KMeans algorithm
training = splitRDD.map(lambda x: array([x[1], x[2]]))
```

Figure 2: Code Snippet Showing Initial Pre-Processing



```
clusters = KMeans.train(training, k, maxIterations=20, initializationMode="random")

# transforms the data into the form
# ((cluster latitude, cluster longitude),
#   [[<month & year>, <event lat>, <event lng>, <religion code>, <quad class>,
#     <goldstein score>, <number of sources>, <number of articles>],
#    [<month & year>, <event lat>, <event lng>, <religion code>, <quad class>,
#     <goldstein score>, <number of sources>, <number of articles>], ...]
# before being mapped to the transform_data function
clusterRDD = splitRDD.map(lambda x: (tuple(clusters.centers[clusters.predict(array([x[1], x[2]]))]), x)) \
                     .groupByKey().mapValues(list).map(transform_data)
```

Figure 3: Code Snippet Showing K-Means Clustering

transformed into JSON format before being saved onto an S3 bucket for further processing and visualisation.

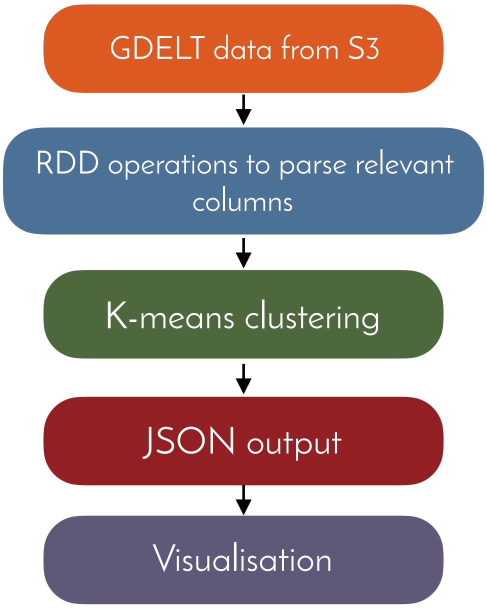


Figure 4: Program Flow

Visualisation

The final processing before visualisation involved finding min-max values for normalisation of data across different months. This normalisation was done using JavaScript during visualisation as we did not wish to increase the complexity of data aggregation on the Spark cluster.

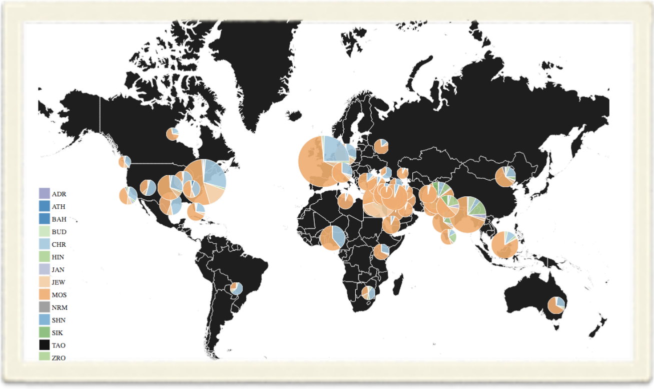


Figure 5: Visualisation of clusters, size of pie chart depicts number of events in May 2016

Visualisation was done using the d3.js library. Pie charts were superimposed upon a map using co-ordinates to show the clusters. Pie charts were chosen in order to visualise the split between the different religions leading to conflict in a region.

The size of the pie chart shows the average intensity of conflicts in that region at that time. The values have been normalised and a minimum value has been given to the clusters to ensure they are visible.

The same visualisation was repeated twice with intensity defined in two different ways: based on number of articles vs based on number of events. This was done in order to show a contrast between the actual impact and media reporting of religious conflicts in different regions.

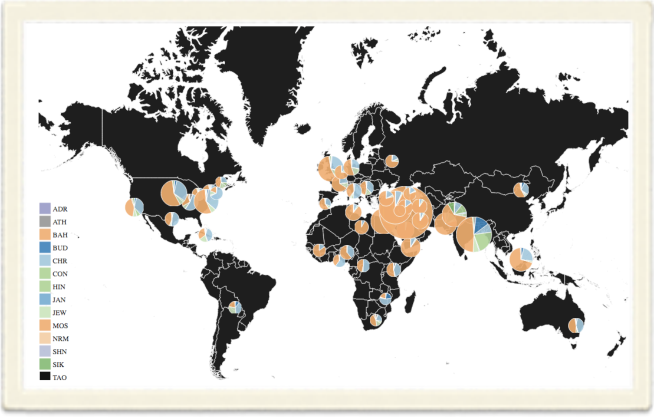


Figure 6: Visualisation of clusters, size of pie chart depicts number of articles in May 2016

Results

We were able to successfully analyse and visualise over 27 GB of data in a little under 2 hours and produced the two visualisations shown in fig. 5 and fig. 6, for 12 months of data from May 2016 to April 2017.

As can be seen in the figures, there is a stark contrast between size of piecharts in the two maps, this shows that in western regions (US and Europe) events tend to be under-reported whereas in the middle east religious conflicts are often over-reported. While the average number of events is about the same in US and Middle East for the given months, there are far more articles about religious conflicts in the middle east.

The distribution of religions remains about the same in the two however, we can notice that there are many instances across the clusters where middle eastern reports will tend to over-represent Muslim conflicts while western reports will under-represent Christian conflicts, this effect is also due to bias in our sources which are global, but widely skewed to feature western media sources.

Additionally, we were also able to identify some interesting, unexpected outliers such as religious conflicts in Hawaii/Alaska, Sikh religious conflicts in Canada and Buddhist religious conflicts in Southern Africa which can also be seen in Fig 6.

Conclusion

Challenges:

The dataset we were working with is extremely large and consists of a wide range of attributes that are irrelevant to our project. Understanding this data before processing it was one of our key challenges.

Setting up the clusters with the correct IAM permissions also took a while and we also had to research the different machine learning algorithms and visualisations libraries that could be used to accomplish our objectives.

Future Work:

Our work has been developed to make limited use of the vast knowledge of the GDELT database due to the limited budget constraints of this project, in the future the same program could be generalised to extract more information for clustering such as ethnicity, daily timing and more to get an even more detailed analysis of religious conflicts.

References

[1] Reuters. "Religious conflict in global rise - report" The Telegraph. The Telegraph Newspaper UK, 14 Jan. 2014. Web. 20 May 2017.

[2] Brian J. Grim. "Religion on the Rise: What this Means for Peace and Conflict" CRG. Religion Freedom & Business Foundation, 02 Apr. 2015. Web. 20 May 2017.

[3] Kalev Leetaru. “Global Database of Events, Language and Tone” GDELT Project. Yahoo! and Georgetown University, Web. 20 May 2017.

Other Resources

- Spark Manual
- GDELT Manual
- Cameo Code Guide