Practical Work

Guilherme Henrique Resende de Andrade

September 2019

1 Introduction

In the last decades, we have noticed the growth of many web applications and, as they increased, also did the amount of data produced by them. Among the big variety of applications, social networks play a key role in producing and consuming data.

In the set of all social networks, when people want to perform any kind of analysis, they always look for the ones capable of providing easy data extraction - ideally through API -, a representative amount of active users, a massive amount of data and some other features such as geographic location, platform, base language, interactions statistics, etc. In such scenarios, Twitter arises as a well-suited application. This practical work was developed over a Twitter dataset with ≈ 2.4 million tweets produced by ≈ 30.000 users.

2 Dataset

The dataset was built by systematically collecting tweets' data from 22-03-2016 to 30-06-2016. The data was downloaded through Twitter API and has a well-defined structure with more than 30 features, where some of them may be objects. For a more formal and complete description of the Tweet object, read the API Reference.

The data extraction process was able to gather ≈ 2.4 million tweets produced by ≈ 30.000 users. The API has a maximum number of requests per day and this limit may vary, the current limit can be found at the API FAQ.

3 Analysis

One of the key aspects people often look for when extracting data from APIs like Twitter's is the structured nature of the data. Well structured data is capable of sparing a substantial amount of preprocessing time, however, it does not eliminates its necessity.

Considering that maintaining all the original features is an expensive process and becomes more difficult over time due to data engineering and feature

extraction steps, this practical work first focus on building a consistent and efficient partial dataset.

3.1 Data Cleaning

The first action performed over the data was to reduce the number of features from 30+ to only 7, namely, coordinates, entities, id_str, created_at, is_quote_status, quoted_status_id_str, text.

Twitter users can quote other's tweets. Originally, a post quoting another post would maintain a reference to the cited one and, considering the quotation is an active part of the message of the current tweet, both the cited and the referrer are concatenated to build only one message.

As shown at the API Reference, every tweet has its timestamp. However, for some unknown reason, all records from this dataset had their timestamps corrupted while converting the data object, i.e. instead of a regular 10-digit variable, it was saved as a 13-digit variable, being the last three numbers zeros. Therefore, to correctly cast these timestamps as DateTime variables, it was first necessary to remove the remaining zeros.

The last steps consisted solely of getting the latitude and longitude coordinates, the hashtags and then extracting the year, month, day and hour from the DateTime variables.

The resulting dataset has the following structure,

```
|-- id_str: string (nullable = true)
|-- text: string (nullable = true)
|-- hashtags: string (nullable = true)
|-- x: float (nullable = true)
|-- y: float (nullable = true)
|-- year: integer (nullable = true)
|-- month: integer (nullable = true)
|-- day: integer (nullable = true)
|-- hour: integer (nullable = true)
```

Figure 1: The dataset final structure

By the end of the cleaning process, it was possible to reduce the original dataset from 7500 MB to 128.5 MB, in other words, a shrinkage to $\approx 1.7\%$ of its original size. The new dataset was persisted as Parquet due to the loading speed, compression, and columnar structure.

3.2 Data Analysis

3.2.1 General Investigation

One of the main forms of improving communication and engagement on Twitter is by using hashtags. In this practical work, we were free to explore any approach, however, it was mandatory to answer two specific questions. The first one regards the discovery of the most frequent hashtag in a given hour through the entire interval. Whereas the second one tries to unveil which moment - Month, Day, and Hour - the 10 most frequent hashtags happened

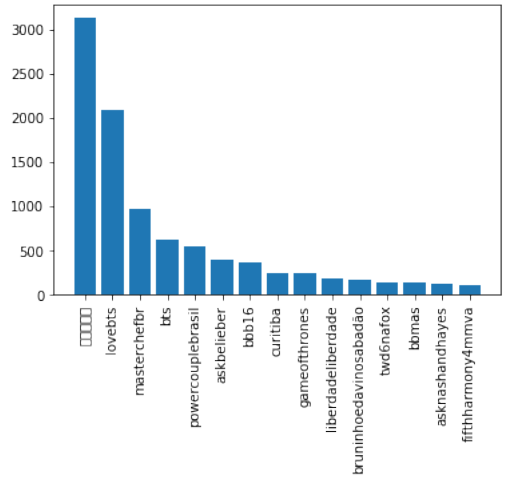

Figure 2: Most popular hashtags.

As we can see above, the most frequent tags were relative to a South Korean band named BTS. However, we are also able to see some other relatively relevant tags such as masterchefbr, powercouplebrasil, gameofthrones, liberdade-liberdade and so on.

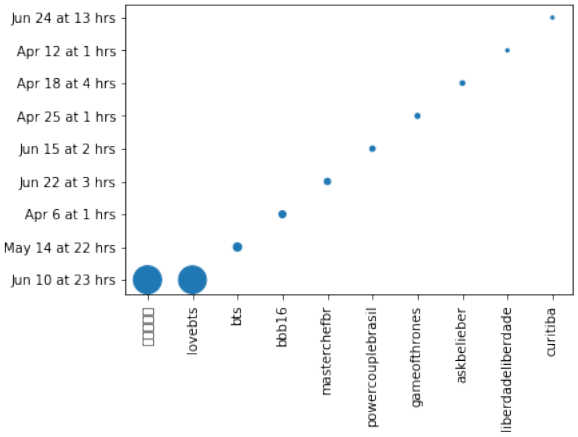

Figure 3: Most frequent date for each hashtag.

Subsequentially, we explored when was the moment in which each hashtag happened the most. In Figure 3, we observe that two out of three BTS hashtags had a use peak on June 10th, 2016. It was not possible - superficially looking on the web - to find if there were any near events and remarkable happenning near this date.

Nonetheless, it is possible to explain some of the remaining hashtags figuring on this list.

- **BBB16**: Had its peak on April 6th, the day after the season finale according to Wikipedia.
- **PowerCoupleBrasil**: Presented an increasing on June 15th, i.e. near the end of the season and, according to O Fuxico, the day which the program reached audience leadership.
- **GameOfThrones**: The most frequent date for this hashtag, according to Wikipedia, was the day after the season premiere, April 24th.

3.2.2 Specific Investigation

The path chosen to be taken was to investigate the geographic properties of the data according to the most frequent hashtags. Hence, firstly it was necessary to perform a clustering based on the coordinate points.

After a quick investigation of the available clustering algorithms for PySpark, we chose K-Means. Given this algorithm has K as a hyperparameter, it was necessary to perform a tunning step. When performing grouping tasks, we must look for both distance error reduction and high consistency of the groupings. Hence, as a way to find the best K, we need to minimize the error rate whereas maximizing the consistency between items in a cluster, the latter is done by using Silhouette.

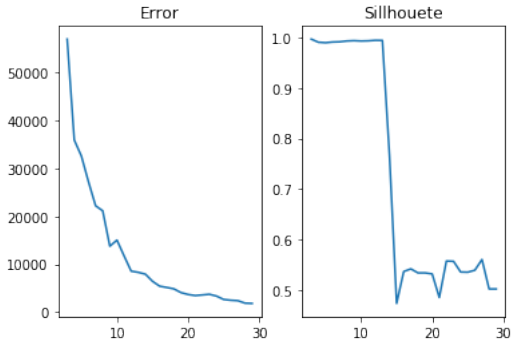

Figure 4: The error curve (left) and the Silhouette curve (right) varying the number of cluster from 3 to 30.

To find the best value for K, we decided to vary this parameter over [3-30]. As we can observe in the chart above, the Silhouette value decays substantially after 12 clusters even though the error continues to decrease. Hence, we define K=12 which can reach 0.974 of Silhouette score.

Since the clustered data was already two dimensional, it was not necessary to perform any kind of dimensionality reduction for visualization. Below we can see how well the clusters can be divided.

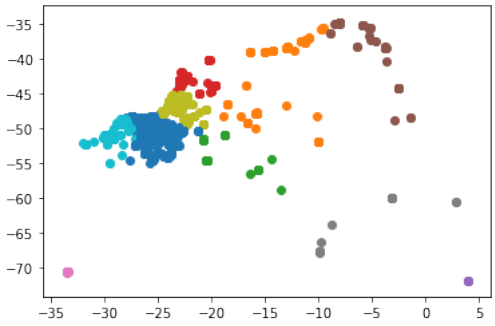

Figure 5: Two dimensional chart with 12 clusters.

Even though this clustering reaches a good consistency over the items of each group, we can see they do not present a well-defined division. The current display indicates that probably fewer clusters would present a more visually appealing grouping.

To validate the fewer groups questioning raised in the last paragraph, we propose a clustering with K=3.

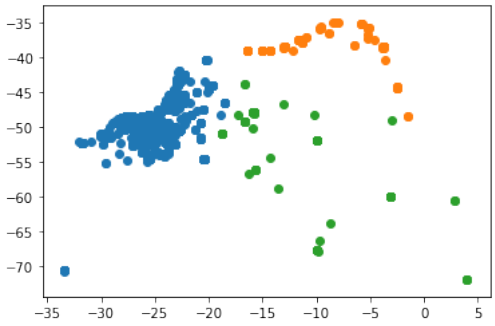

Figure 6: Two dimensional chart with 3 clusters.

Indeed, the plot above presents a more reasonable grouping, even though, as shown in 4, this parameter presents the biggest error rate. However, for comparison purposes, the plots bellow will also present, alongside K=12, K=3 (biggest error rate) and K=30 (smallest error rate) clusterings.

Initially, we can see the outline of South America (Figure 3.2.2) and, consequently, we notice that, contrary to what was thought, the dataset does not comprise only tweets from Curitiba. As the number of clusters increase, it is evidenced that the dataset has, in reality, a wider range of places committing posts.

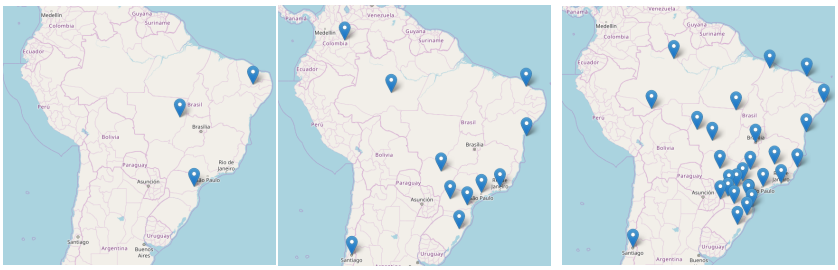

Figure 7: Geographic chart with pins representing the cluster center (centroid). From left to right we are able to see the outline of South America from executions with K=3, K=12 and K=30.

Below we present the delineation of Curitiba as a way to better understand what happens to the city's centroids as we increase the number of clusters.

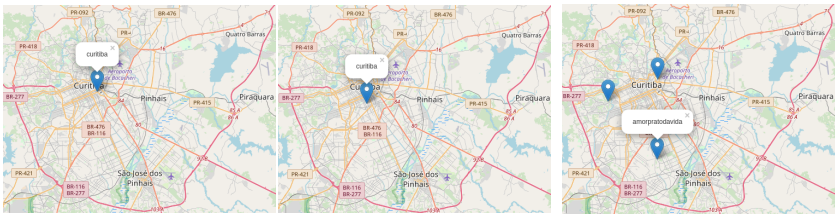

Figure 8: Geographic chart with pins representing the cluster center (centroid). From left to right we are able to see the outline from Curitiba (in red) with the most frequent hashtag of each cluster with K=3, K=12 and K=30.

Figure 3.2.2 presents three different K-Means executions with K equal to 3, 12 and 30, respectively. The blue pin in the image represents the geographic cluster center and, the label above the pin is the most frequent hashtags in the specific cluster.

One of the key points in Figure 3.2.2 is that, as we increase the number of clusters, we become able to notice the differentiation of new groups, with even different most frequent hashtags.

4 Conclusion

Through this practical work, we saw that preprocessing the data, even when already structured, to our specific needs can improve the quality of the analysis in both performance and complexity. It was also shown the possibility of concluding out of the data when we combine the original data with additional information from other sources such as web pages, social media, etc.

With the geographic exploration we were able to see that, even though the original dataset was meant to only have tweets from Curitiba, it has indeed post from a wide variety of places.

It was also possible to notice that the clusters' separation had almost always both geographic and topic differentiation. The third image in Figure 3.2.2 shows that even cluster inside Curitiba have different main topics.

Another idea of what could be explored with the dataset include topic modeling per region, performing a sentiment analysis over the tweets and try to investigate whether more frequent users tend to post more sad and depressive content, etc.

5 References

1. https://en.wikipedia.org/wiki/Apache_Parquet
2. https://spark.apache.org/docs/2.1.0/api/python/pyspark.sql.html
3. https://python-visualization.github.io/folium/
4. https://spark.apache.org/docs/2.0.0/api/python/pyspark.mllib.html