Final Report on Application-Focused Project using GDELT 2.0 Data Set

DATA301
Blue Le - 41086772
Kai Koh - 64276083
Group: Data Freaks

Throughout the course of the project, there were several setbacks that hindered our progress, which will be further elaborated in section 1. Therefore, we have decided to modify the research question into “Has Marvel’s popularity increased during the month of April in anticipation of Avengers: Endgame’s release?” to fit GDELT’s limitations. In order to achieve this, the project will make use of GDELT 2.0 GKG data subset regarding Marvel movies and analyse the harvested data using Locality Sensitive Hashing algorithm. The expected result is that the popularity will slowly increase as the release date of Avengers: Endgame grows closer, before sky-rocketing during the few days of its release. This result is significant as it shows how a community that have been built around comics, fictional superheroes and worlds can spark huge global interests, especially with the widely-anticipated Avengers: Endgame concluding the 22-movie, decade-long saga of the Marvel Cinematic Universe. Also, it can also prove how great of an influence Marvel has on the movie industry.

1 Introduction

1.1 Background

To understand what type of GDELT 2.0 database subsets we will be using, we first need to know the theme, timeframe and how the data harvested can support our hypothesis. Furthermore, in order to understand which algorithm may be best-suited to address our research question, we have to look at the type of data, the size of it and how we are going to search for similar articles in our field of interest.

1.2 Research Question

Our question is “Has Marvel’s popularity increased during the month of April in anticipation of Avengers: Endgame’s release?”, which had initially been “How much has Marvel’s popularity increased since its first Phase 1 movie Iron Man, 2008?”. The key factors behind this change in direction are firstly, due to the limitations imposed by GDELT that prevented us from looking up data earlier than 2017. Secondly, another major setback is while accessing and retrieving data from GDELT, a lot of system resources and time are spent compiling these datasets. Thus, in order to work within the timeframe of the project, we have decided to remodel our research question. We expect that the GDELT 2.0 GKG will be most suited since as our research topic is related to media and entertainment. By implementing the LSH algorithm, we can search through millions of records efficiently to sieve out articles related to Marvel Studios and its movies.

2 Experimental Design and Methods

2.1 Group and Individual Taskings

After setting up PySpark and importing GDELT 2.0 into the Google Colab sheet, we initialised the helper functions and dictionary containing the vocabulary of all Marvel-related words. Next, we proceeded to download and write the GKG data from the period of 1 April 2019 to 1 May 2019, as that was the period of time where Avengers: Endgame came to the big screens. The program then inputs these documents into the LSH system: Shingling, Min-Hashing then Locality Sensitive Hashing to produce articles related to our field of interest.

We used Google Docs to maintain the writing for our report, and Google Colab so that we had instant access to the code even when we worked from home. For communication, we kept each other updated via Facebook's Messenger to discuss and arrange meetings for the project. Kai was in charge of compiling the bag of words containing search terms that relate to the Marvel movies as well as implementing the Minimum Hash algorithm. Blue took care of inputting GDELT data into the programme and implementing the Locality Sensitive Hashing. Finally, we combined our individual parts to obtain the extract relevant data from GDELT and proceeded to analyse the findings of our research.

2.2 Coding Implementation

We have created a huge bag of words that had been compiled with words related to Marvel's Cinematic Universe movies. This includes the cast, background crew, movie characters as well as movie titles. As we were unable to set a range of dates to automate the retrieval of the GKG data, we added a retrieve_data() function to address this issue. Helper functions like dbg(x) and load_RDD have also been included for debugging and loading RDDs respectively. Thereon, the filter_documents(filename) sieves out relevant articles, which are then computed for Jaccard Similarity using the jaccard_similarity(list1, list2) function between each document and the dictionary of Marvel-related vocabulary words. We also displayed this result using pandaframe and pyspark.sql's SQLContext. Finally, we processed the data points using the minhash.py, lsh.py modules to obtain the trend results.

3 Results and Analysis

3.1 Results

Unfortunately, we were unable to correctly implement our individual parts together as the final LSH module was not able to accept our input data points. We ran into an error of - java.lang.OutOfMemoryError: Java heap space - while running our main LSH module as seen in Figure 1. Furthermore, the nature of GDELT data consumed a lot of resources and time such that it made the debugging process very slow and staggered.

```
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$abortStage$1.apply(DAGScheduler.scala:1876)
        at scala.collection.mutable.ResizableArray$class.foreach(ResizableArray.scala:59)
        at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:48)
        at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.scala:1876)
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:926)
        at org.apache.spark.scheduler.DAGScheduler$$anonfun$handleTaskSetFailed$1.apply(DAGScheduler.scala:926)
        at scala.Option.foreach(Option.scala:257)
        at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGScheduler.scala:926)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceive(DAGScheduler.scala:2110)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:2059)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive(DAGScheduler.scala:2048)
        at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:49)
        at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.scala:737)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2061)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2082)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2101)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2126)
        at org.apache.spark.rdd.RDD$$anonfun$collect$1.apply(RDD.scala:945)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:151)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScope.scala:112)
        at org.apache.spark.rdd.RDD.withScope(RDD.scala:363)
        at org.apache.spark.rdd.RDD.collect(RDD.scala:944)
        at org.apache.spark.api.python.PythonRDD$.collectAndServe(PythonRDD.scala:166)
        at org.apache.spark.api.python.PythonRDD.collectAndServe(PythonRDD.scala)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:357)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:132)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.GatewayConnection.run(GatewayConnection.java:238)
        at java.lang.Thread.run(Thread.java:748)
Caused by: java.lang.OutOfMemoryError: Java heap space
```

Figure 1: Out of Memory Error

3.2 Analysis

Figure 2: Marvel's popularity from April 2019 - May 2019

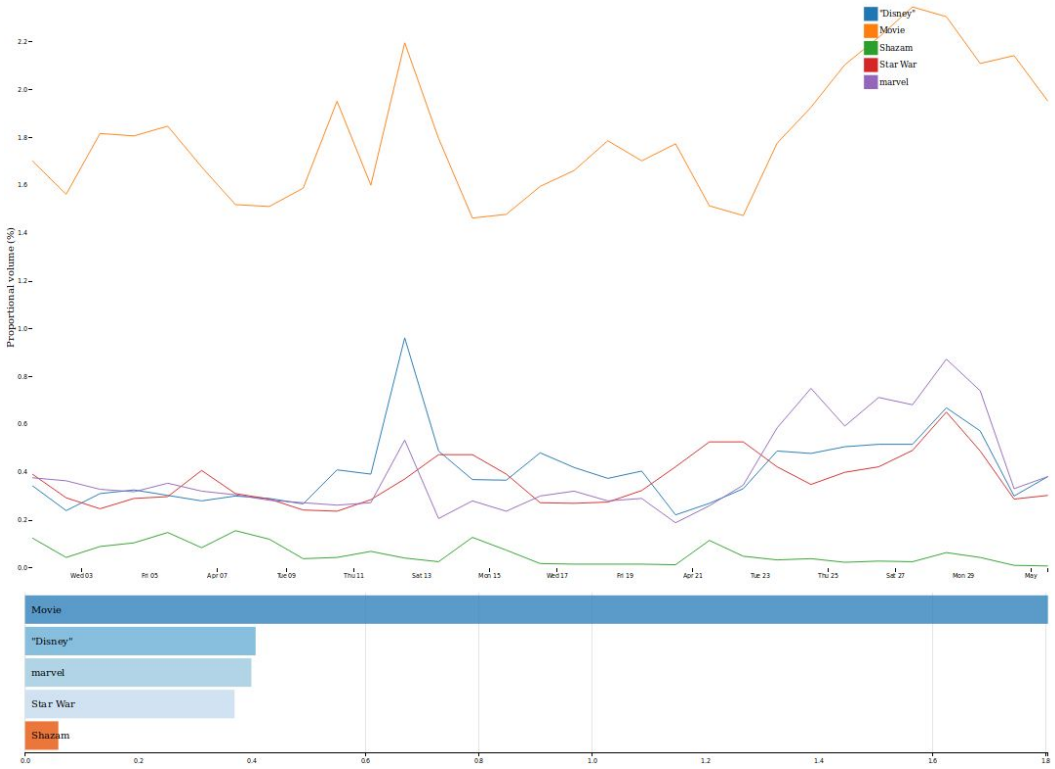

Figure 3: Marvel when compared to other movies and movie companies

As such, we looked for alternatives to represent and visualise the relevant data set, and managed to do that through https://gdelt.github.io/. This allowed us to query for Marvel-related news and articles during the month of April 2019.

Figure 2 showed us the massive success of Marvel's latest blockbuster, Avengers: Endgame. We can clearly see the rippling effect that it has on the entire movie industry as well. The spike in popularity near 13 April 2019 occurred due to Marvel launching its final trailer for the movie. As the movie was released, interest and popularity levels soared steeply. However, we noticed that it quickly died down towards the end of the month.

Based on Figure 3, we can see that when compared to other movies companies, Marvel have clearly taken the lead, even after the hype has died down towards the end of the month; still maintaining at levels higher than other movie companies. Also, we can infer from the chart that Marvel has a high influence on the movie industry. When their popularity rose, not only did Disney - their partner - saw an increased interest, but also the general movie industry as a whole. Their rival - DC Entertainment - had also

released “Shazam” during this period. However, its popularity level is nowhere near Marvel’s, as evident from Figure 3.

4 Conclusion

Based on the results above, we can then answer the research question of “Has Marvel’s popularity increased during the month of April in anticipation of Avengers: Endgame’s release?”. Yes, Marvel’s popularity has increased during the month of April, albeit dying down towards the end of the month leading into May. From the results, we can tell that Marvel also has a huge impact on the movie industry, as its swing in popularity levels can affect the entire industry as seen in the Figure 3 from section 3.

5 References

https://github.com/aamend/spark-gdelt

https://github.com/velvia/spark-sql-gdelt

https://linwoodc3.github.io/gdeltPyR/
https://blog.gdeltproject.org/gdelt-2-0-our-global-world-in-realtime/
http://www.mmds.org

https://www.forbes.com/sites/kalevleetaru/2016/01/13/mapping-world-happiness-and-conflict-through-global-news-and-image-mining/#66f3263ce224
https://en.wikipedia.org/wiki/List_of_Marvel_Cinematic_Universe_films
https://en.wikipedia.org/wiki/List_of_Marvel_Cinematic_Universe_film_actors
https://gdelt.github.io/
https://towardsdatascience.com/overview-of-text-similarity-metrics-3397c4601f50