Project report

Calin Iaru(s1077000)

June 2023

Description and motivation

The focus of this project is to compare the references to 3 of the most famous and used social media platforms: Facebook, Instagram, and TikTok. Analyzing which of these platforms is more present on different websites can help to see which one is the most popular one during the period when the data was archived. Doing this process over different periods of time can tell which platforms are the most influential.

Facebook and Instagram are apps used for over one decade, but TikTok is quite new but is gaining popularity quite fast, in 2021 being downloaded over 3 billion times. Having this in mind, I thought that analyzing references to this app and comparing it with the other 2 most famous social media apps would be interesting, especially since I consider only references to these 3 apps and compared tier percentages.

Process

I started by defining some smaller functions that can help me to achieve my goal.

The first step was to use the Jsoup library to parse the HTML content of each web page contained in the WARC files. After a file is translated to an HTML page, the function `extractUrls` saves all the links that are inside an object of type `<a>` that has as an attribute: `href` which is used to denote a reference to a website. The next function that I used is a simpler one, `containsReference`, it only checks if a given domains appear in the list of links stored by the previous function.

The main function of the code, `references` takes a DataFrame called `warcData`. This DataFrame is crafted by combining the first 100 WARC files in the folder `hdfs:///single-warc-segment/`. For each line in `warcData` the following steps are performed:

1. search for the index of "WARC-Type", this step is necessary, as not all the lines in the WARC data contain this identifier and they are not useful and there is no need to evaluate these lines;
2. if we found an index, the white spaces are removed, this helps in finding the exact type of the WARC record(e.g. response, request), as we are interested only in the response type as there we can find the references;
3. when getting a response record, the code search for the index where the body starts and check if the body actually contains something;
4. the function `extractUrls` is called and stores all the references to other websites.

If any of the previous steps fail, then an empty list is returned.

The next of the code is based on searching the references DataSet for references to the 3 social media platforms and categorizing them by `facebook`, `instagram`, `tiktok` and `others` if the references are not referring to any the 3 social media apps, and also count them. Then filtering is done to store only the desired references and finally, they are compared by their percentage.

I copied my code to the `RUBigDataApp.scala` file and saved on the cluster. Then I created a .jar file using the `sbt assembly` command, the necessary files with all the dependencies and then I submitted to the cluster.

Testing and challenges

For testing, I started by just taking one file of the segment list, and then gradually I increased the number of files from 1 to 10 to 50 and finally to 100, I think I could've run it on also more files and still gotten a SUCCEEDED.

Some challenges that I encountered were correctly identifying when the body of the response starts, how to get the actual index of the body, and I was constantly forgetting about the possibility that my reference could be empty and not returning something in that case.