Big Data Project Report

Daniel Matias Suryadi Putra

S1078257

Topic: Finding the most frequently used word in the HTML body of the NIKE SNKRS WARC File

Description of the Basic Idea:

This topic was inspired by the idea of exploring words beyond the commonly used ones listed on Wikipedia, we can filter out those words to obtain a top-frequent list that may be more interesting and relevant to the NIKE SNKRS website, such as words like "nike" and others related to the footwear industry.

The basic idea behind this project is to analyze the content of the NIKE SNKRS WARC file by focusing on the HTML body and identifying the most frequently used word. By extracting this information, we can gain insights into the website's content. This project involves processing the HTML body information, which is explained below.

The Process:

The process of this project goes as follows:

1. Data exploration
    a. Reading the WARC file
    b. Understanding the WARC file schema to identify relevant columns
2. Data extraction
    a. Extracting Header information, including the content language, to gather additional insights
    b. Extract information on HTML body from the HTML content using Jsoup
    c. Count the occurrence of the words that is used:
        i. Tokenizing each sentence by splitting it into individual words
        ii. Count the occurrence of each word using wordcount
    d. Show/identify the top 10 most frequently occurring words based on the wordcount
        i. Also don’t forget to filter out curly braces and numbers
3. Uploading it/ Making it into a standalone application
    a. Copy paste and modify the code to the RUBigDataApp.scala file in a text editor
    b. Build the stand-alone application using “sbt assembly” because of the jsoup that I use
    c. Submit the stand-alone application to the cluster

Divide and Conquer:

For the project, the things, and steps that I have done is documented as below:

1. Explore the NIKE SNKRS website WARC file and understood its structure
2. Extracted header information, including content language, to analyze language-related insights (not used at the final product).
3. Use Jsoup to extract the HTML body from the website's HTML content
4. Tokenized the HTML body to separate words and counted their occurrence
5. Filter only a-z and A-Z
6. Identify the top 5 most frequently occurring words in the HTML body using wordcount
7. After everything is finished, uploading it to the cluster is the last step by making it first into the stand-alone application
    a. Modified some code so that it works in the stand-alone-application

WARC File details:

The WARC file I use was based on the website https://www.nike.com/nl/launch from Nike SNKRS.

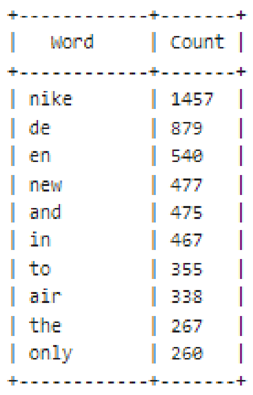
One change from the zeppelin code and the stand-alone application code is that in the zeppelin code I use:
val warcfile = "nike.warc.gz"

Since its at my big data, but now I must change it into:
val warcfile = "hdfs:///user/s1078257/nike.warc.gz"

This is so that it works on the cluster.

Result:

So, after performing the task that have been mentioned from the divide and conquer steps and the process, I have managed to get the top 10 most occurring word in the HTML body of the Nike SNKRS WARC file which is as following:

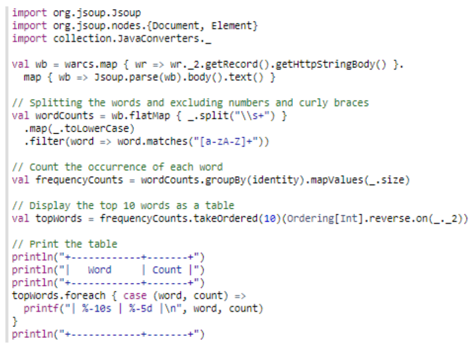


```
+------------+-------+
|   Word     | Count |
+------------+-------+
| nike       | 1457  |
| de         | 879   |
| en         | 540   |
| new        | 477   |
| and        | 475   |
| in         | 467   |
| to         | 355   |
| air        | 338   |
| the        | 267   |
| only       | 260   |
+------------+-------+
```

From the following output, it can be seen that “nike”, which is the brand have the most occurrence here.

What went right:

Most of things that I have done went right. I have done the wordcount and tokenizing each sentence successfully from the HTML body which successfully give the top 10 most occurring word. I used Jsoup that is given in the WARC to Spark file that is given and modify it so that it gives content of the HTML body rather than the header. Then the tokenizing is done with flatmap and the wordcount is done with grouping as can be seen from the code documentation below.



```
import org.jsoup.Jsoup
import org.jsoup.nodes.{Document, Element}
import collection.JavaConverters._

val wb = warcs.map { wr => wr._2.getRecord().getHttpStringBody() }.
  map { wb => Jsoup.parse(wb).body().text() }

// Splitting the words and excluding numbers and curly braces
val wordCounts = wb.flatMap { _.split("\\s+") }
  .map(_.toLowerCase)
  .filter(word => word.matches("[a-zA-Z]+"))

// Count the occurrence of each word
val frequencyCounts = wordCounts.groupBy(identity).mapValues(_.size)

// Display the top 10 words as a table
val topWords = frequencyCounts.takeOrdered(10)(Ordering[Int].reverse.on(_._2))

// Print the table
println("+------------+-------+")
println("|   Word     | Count |")
println("+------------+-------+")
topWords.foreach { case (word, count) =>
  printf("| %-10s | %-5d |\n", word, count)
}
println("+------------+-------+")
```

I personally think that this part took the longest time because I got a lot of difficulty on getting information for the HTML body since before it only gives information on the title and URLs but after some trial and eror, it finally works.

Some issue which have been fixed:

Even though a lot of things went right, there are some things that also I have some difficulty too. One of which is extracting the Content-language from the header. My first idea was to identify the most frequently language used in this website from the header. However, this went wrong for some reason and all the language was given as “Unknown”. With this beeing a mistake, then I have decided to change the topic into extracting the HTML body and doing wordcounts on it since there are more content in the HTML body that can be analyzed rather than the small content of the header which does not provide enough information. Before the HTML body, my first idea was to extract just the header, but I thought that there wouldn’t a lot of interesting stuff there since its not as big as the HTML body.

Conclusion:

In conclusion, the analysis of the NIKE SNKRS website's HTML body using the provided WARC file allowed us to identify the most frequently used word. This analysis provided insights into the website's content and patterns.

Through data exploration and extraction, we successfully obtained the HTML body from the WARC file and counted the occurrence of words. By tokenizing the HTML body and performing word count analysis, we determined the top 10 most occurring words.

Although there were some challenges, such as extracting the content language from the header which absolutely does not work and adapting the code for standalone application submission, the project was largely successful.