WARM-UP PROJECT (NEW YORK TIMES)

Affan Asad Charolia
Aqdus Arshad Charolia
Xuanchang Ye

OBJECTIVE

The objective of this project is to use the New York Times articles data and perform various tasks on it, which includes:

- Task 1: Build a word cloud for NY Times articles using Apache Hadoop or spark and list top 100 words used in all articles.
- Task 2: Word count for top 5 news category.
- Task 3: List the top 10 words that are shared among the highest number of news articles of the same category.

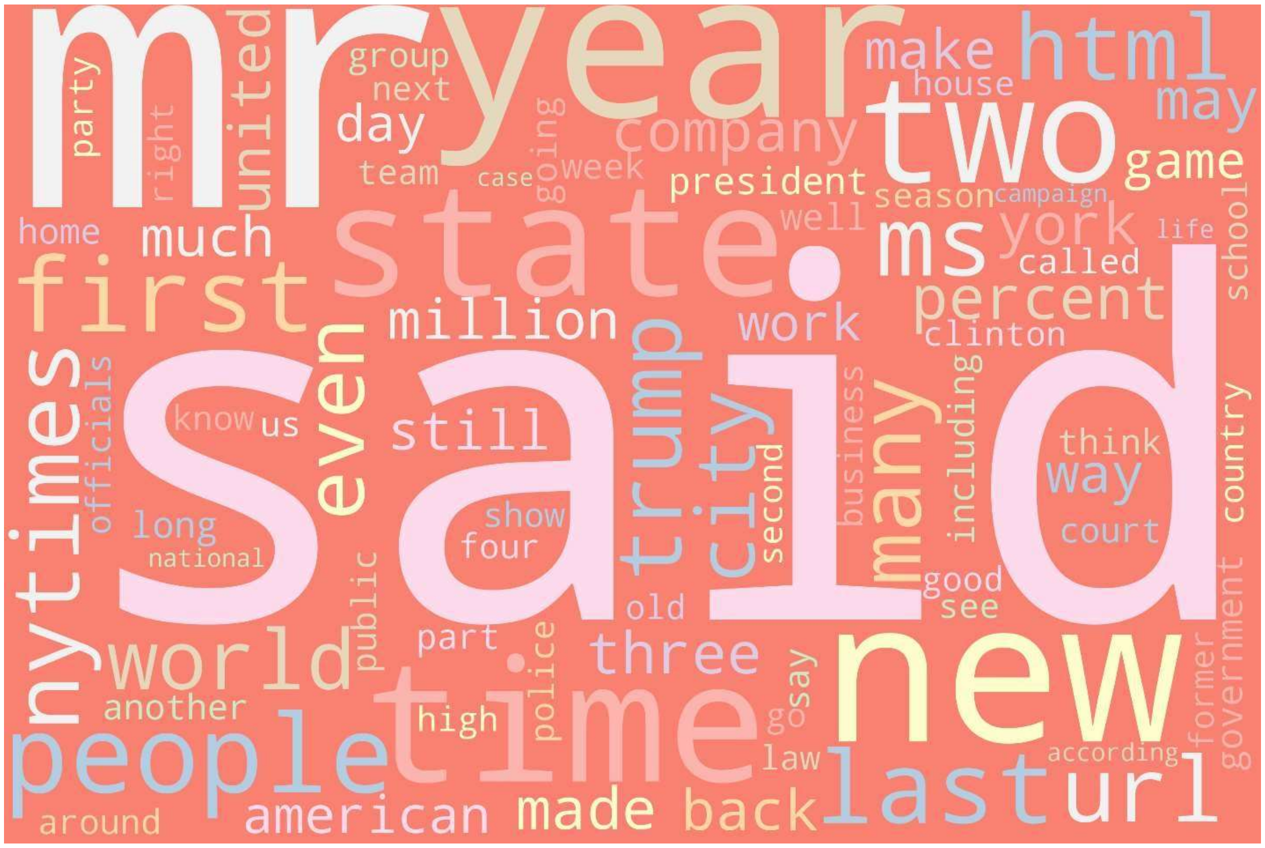
METHOD USED

The project was executed on Google Colab using Apache Spark version 3.0.2. The dataset file and all the intermediate files created were stored in Google Drive. Word Count was done using the help of Apache Spark, and python library 'pyspark'. Other important python libraries used were 'nltk' for removing punctuations, 're' for regular expression, and 'matplotlib', 'wordcloud' for plotting the Word-cloud.

TASK 1

- The dataset is imported from the google drive and all the punctuations are removed and the output is stored in an intermediate file.
- Spark is loaded in system and the MapReduce function is performed on the intermediate file.
- Before performing Word Count on the data all the stop words are removed using *'splitRDD_no_stop = words.filter(lambda x: x.lower() not in stop_words)'* command.
- The Word Count is then performed on the data which returns words as key and their count as value.
- The words are then sorted on basis of their count in the descending order.
- The word and count of the top 100 words are stored in the 'out_task1' file.
- The word cloud is made for the top 100 words using the 'out_task1' file.

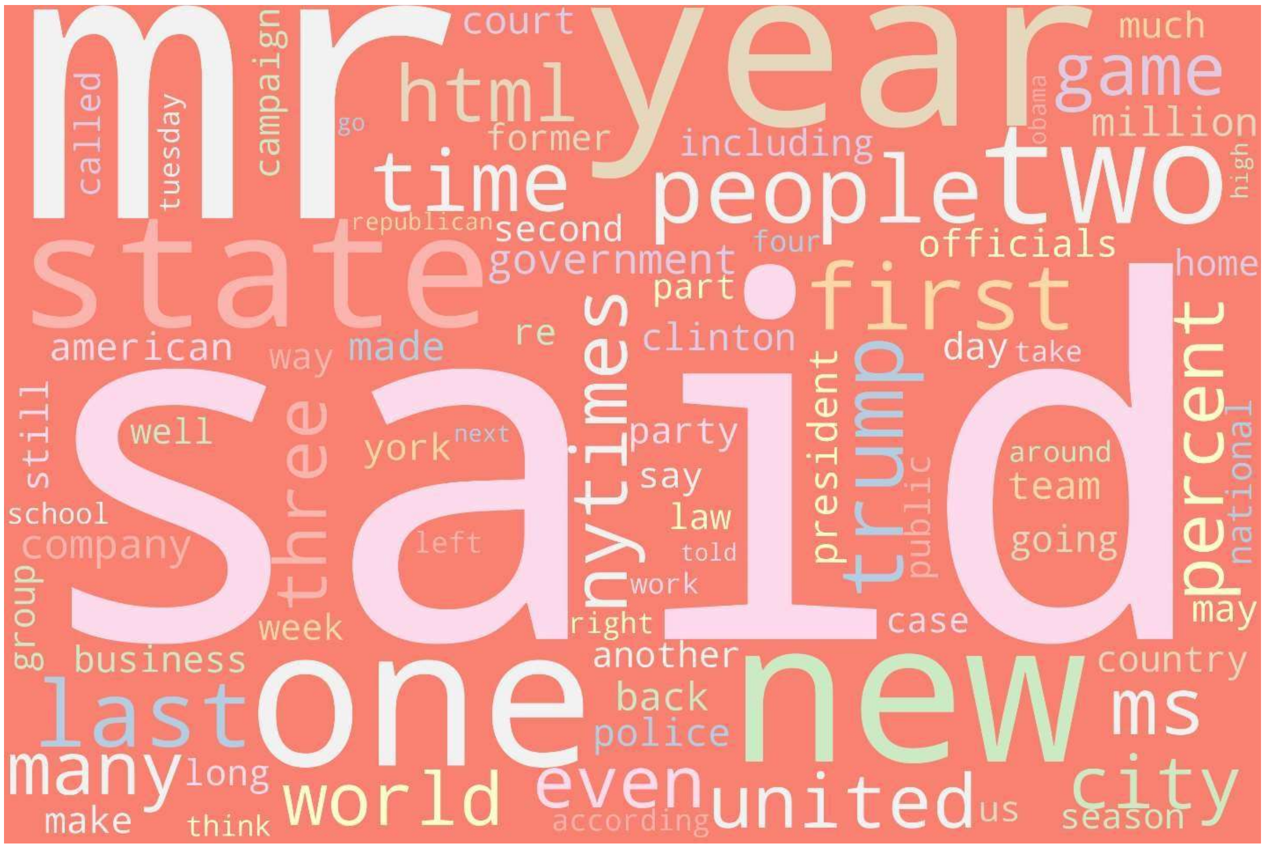
Output- Word Cloud for top 100 words found in the news article.



TASK 2

- The dataset is imported from the google drive.
- URLs are extracted from the file using regular expression *'(https?://\S+)'* and storing it in *'urls'* file.
- Extracting category names from the URLs using regular expression *'://[\w\-\.]+/[0-9]+/[0-9]+/[0-9]+/([\w\-]+)'*.
- Starting the spark session.
- Doing word count using MapReduce and finding top 5 news categories by sorting it.
- Importing the nytimes files and splitting the entire file at 'url:'.
- Adding the articles of top 5 news category to file *'out_with_topfive_category1'*.
- Removing punctuations from the file.
- Removing Stop Words and performing word count using MapReduce.
- Taking top 100 words and storing it in *'out_task2'* file.
- The word cloud is made for the top 100 words using the *'out_task2'* file.

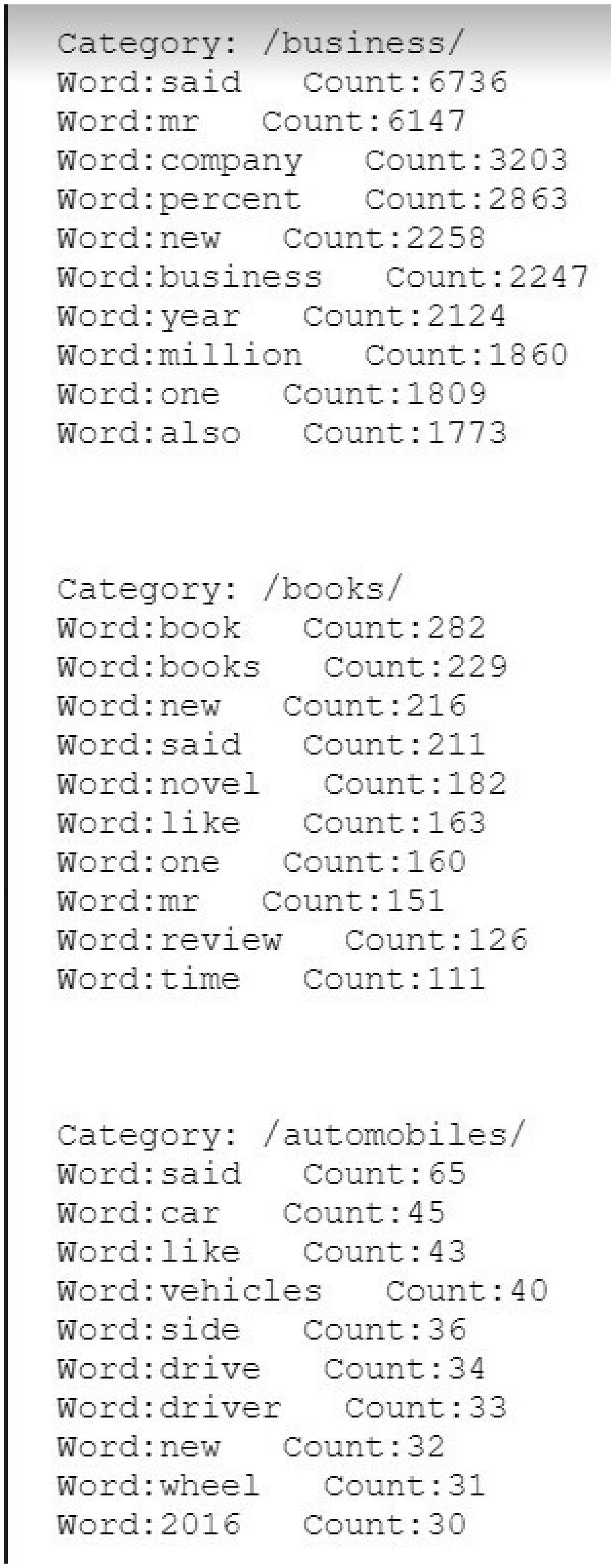
Output- Word Cloud for the top 100 words from articles of the top 5 news categories.



TASK 3

- Starting the spark session.
- Extracting all the category names from the URLs using regular expression *'://[\w\-\.]+/[0-9]+/[0-9]+/[0-9]+/([\w\-]+)'*.
- Importing the nytimes files and splitting the entire file at 'url:'.
- Running a 'for' loop for each category:
    - Adding all the news articles of that category in the list.
    - Removing Punctuations.
    - Performing word count using Map Reduce.
    - Forming list for top 10 words.
    - Adding the category name and the word and count for top 10 words.
- Saving the output in *'out_task3'* file.

Output- Screenshot of top 10 words for each news category.



```
Category: /business/
Word:said   Count:6736
Word:mr   Count:6147
Word:company   Count:3203
Word:percent   Count:2863
Word:new   Count:2258
Word:business   Count:2247
Word:year   Count:2124
Word:million   Count:1860
Word:one   Count:1809
Word:also   Count:1773



Category: /books/
Word:book   Count:282
Word:books   Count:229
Word:new   Count:216
Word:said   Count:211
Word:novel   Count:182
Word:like   Count:163
Word:one   Count:160
Word:mr   Count:151
Word:review   Count:126
Word:time   Count:111



Category: /automobiles/
Word:said   Count:65
Word:car   Count:45
Word:like   Count:43
Word:vehicles   Count:40
Word:side   Count:36
Word:drive   Count:34
Word:driver   Count:33
Word:new   Count:32
Word:wheel   Count:31
Word:2016   Count:30
```