Big Data Analytics Project-ITC 686

Google ngrams

Central Michigan University

Dept. of Computer Science-Spring 2016

Project Members: Rakasi Sandeep Reddy (rakas1s@cmich.edu)

Data Source: http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

Data Set Description: The data mainly consists of five fields. They are

! ! ! that	1800	2	2	2
! ! ! that	1810	3	3	3
! ! ! that	1812	1	1	1

The data format is as follows:

Ngram TAB year TAB match_count TAB page_count TAB volume_count NEWLINE

The first line in the data set tells us that in 1800, the word "! ! ! that "occurred 2 times overall, on 2 pages in 2 distinct books of our sample.

Each of the files below is compressed *tab*-separated data.

Data pre-processing: the data downloaded was in zip format which was of 300Mb each.

There were many files of this type. So I planned of combining them all and unzipping them.

Unzip /home/rakas1s/project/data/raw/* -d /home/rakas1s/project/data/raw/

Then I combined all the files in to one large file which is of 190 GB and deleted all the small files

cat *.csv >> ngrams.csv

Schema Checking:

The data collected may have some null values or different data types. So checked the schema using a shell script.

1. First column is checked for only null values as it is a string a string and can have any type of data in that

2. All the remaining columns are checked for null values and integers

Data Ingestion:

Shell Script to check the schema and place data onto HDFS:

```
#Description: This script with do schema validation for each column of "ngrams.csv" file and will parse
through only matched columns into final file. Script first starts with checking the existence of the
original source file and will only proceed if file is present. Once the Schema Validation is done, the file
is automatically placed on HDFS. The Script shows message at each and every step of the execution.
#Global-Id: rakas1s
#Creation Date: 04/24/2016
#Version: 1.1

echo "Info: Initiating Script....."
sleep 2

date1=`date +%m-%d-%y`
time1=`date +%T`
file_name=ngrams.csv

echo "Info: Script Started on $date1 at $time1"

src_path=/home/rakas1s/project/data/raw/
hdfs_path=/user/rakas1s/project/data/raw

echo "Info: Changing Directory to $src_path"
sleep 2

echo "Info: Checking if $file_name is present under path $src_path"
sleep 2

if [ ! -f $file_name ]
```

```
	then
		echo "Error: File $file_name doesn't exist under path $src_path. Exiting Script."
		exit 1
	else
		echo "Info: $file_name File Exists!"
fi

echo "Info: Initiating Schema Check"
sleep 2
awk -F"\t" '$1>"  " {print $0}' $file_name > ngrams1.csv
echo "Info: Column1 Parsed Successfully"

awk -F"\t" '$2>"  " {print $0}' ngrams1.csv | awk -F"\t" '$2~ "^[0-9][0-9]*$" { print $0 }' > ngrams2.csv
echo "Info: Column2 Parsed Successfully"
echo "Info: Removing ngrams1.csv"
rm -f ngrams1.csv
echo "Info: ngrams1.csv Removed Successfully"

awk -F"\t" '$3>"  " {print $0}' ngrams2.csv | awk -F"\t" '$3~ "^[0-9][0-9]*$" { print $0 }' > ngrams3.csv
echo "Info: Column3 Parsed Successfully"
echo "Info: Removing ngrams2.csv"
rm -f ngrams2.csv
echo "Info: ngrams2.csv Removed Successfully"

awk -F"\t" '$4>"  " {print $0}' ngrams3.csv | awk -F"\t" '$4~ "^[0-9][0-9]*$" { print $0 }' > ngrams4.csv
echo "Info: Column4 Parsed Successfully"
echo "Info: Removing ngrams3.csv"
rm -f ngrams3.csv
echo "Info: ngrams3.csv Removed Successfully"
```

```
awk -F"\t" '$5>"  " {print $0}' ngrams4.csv | awk -F"\t" '$5~ "^[0-9][0-9]*$" { print $0 }' > ngrams5.csv
echo "Info: Column5 Parsed Successfully"
echo "Info: Removing ngrams4.csv"
rm -f ngrams4.csv
echo "Info: ngrams4.csv Removed Successfully"

sleep 2
echo "Info: Schema Check Completed Successfully"

echo "Info: Renaming ngrams5.csv to $file_name"
echo "Warning: Old file will be overwritten"
mv ngrams5.csv $file_name
sleep 2
echo "Info: Moving Files on HDFS"

hadoop fs -copyFromLocal -f $src_path$filename $hdfs_path
echo "Info: $file_name Moved Successfully on HDFS path $hdfs_path"

echo "Info: Exiting Script"
```

Batch Layer:

Tool used to prepare batch_views: hive

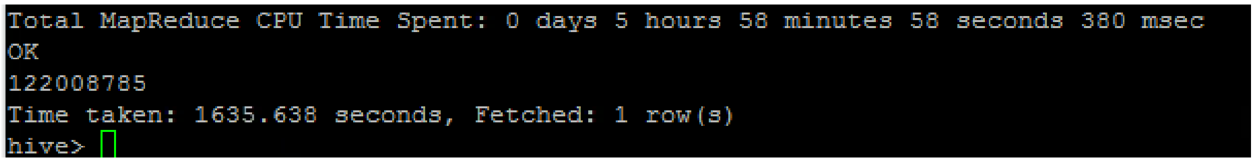
Creation of table using hive:

```
CREATE EXTERNAL TABLE ngrams (

ngram STRING,

year INT,

match_count INT,

page_count INT,

volume_count INT

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY "\t"

STORED AS TEXTFILE

LOCATION '/user/rakas1s/project/data/raw/';
```

Queries:

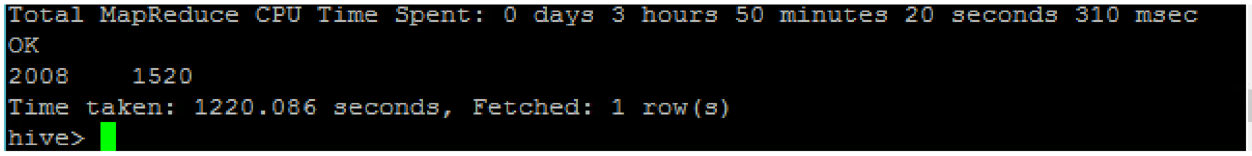
Executed some basic queries to find some basic information.

SELECT COUNT (B) FROM (SELECT distinct (ngram) AS B FROM ngrams) A;



```
Total MapReduce CPU Time Spent: 0 days 5 hours 58 minutes 58 seconds 380 msec
OK
122008785
Time taken: 1635.638 seconds, Fetched: 1 row(s)
hive>
```

SELECT MAX (year), MIN (year) FROM ngrams;



```
Total MapReduce CPU Time Spent: 0 days 3 hours 50 minutes 20 seconds 310 msec
OK
2008    1520
Time taken: 1220.086 seconds, Fetched: 1 row(s)
hive>
```

All the above commands are present in *queries.hql* under the path */home/rakas1s/*

Batch views:

Batch view is created to reduce the latency of results to a query, as operation on a batch view need not go through complete data set. We only separate the data that we need to query and create a batch view.

Batch view 1:

This batch view is generated based on the year 1520 and 1700

hive -e select ngram,MAX(match_count) as max_match,MIN(match_count) as min_match from ngrams where year between 1520 and 1700 group by ngram > /home/rakas1s/project/batch_views/batch_view1.csv'

Piping files to provide it to Serving layer for further processing:

cat project/batch_views/batch_view1.csv | sed -e 's/\t/,/g' > project/batch_views/batchview_a.csv

cat project/batch_views/batchview_a.csv | sed -e 's/"//g' | sed -e 's/,,/,/g' >
project/batch_views/batchview_b.csv

rm -f project/batch_views/batchview_a.csv

cat project/batch_views/batchview_b.csv | awk -F ',' '{printf(%s,%i,%i,\n,$1,$2,$3)}'\ >
project/batch_views/batchview_c.csv

rm -f project/batch_views/batchview_b.csv

mv project/batch_views/batchview_c.csv project/batch_views/batch_view1.csv

mv batch_view1.csv batch_view1.sql

all these above shell commands are placed in script *batch_view.sh* under path /home/rakas1s/project/batch_views/

Batch view 2:

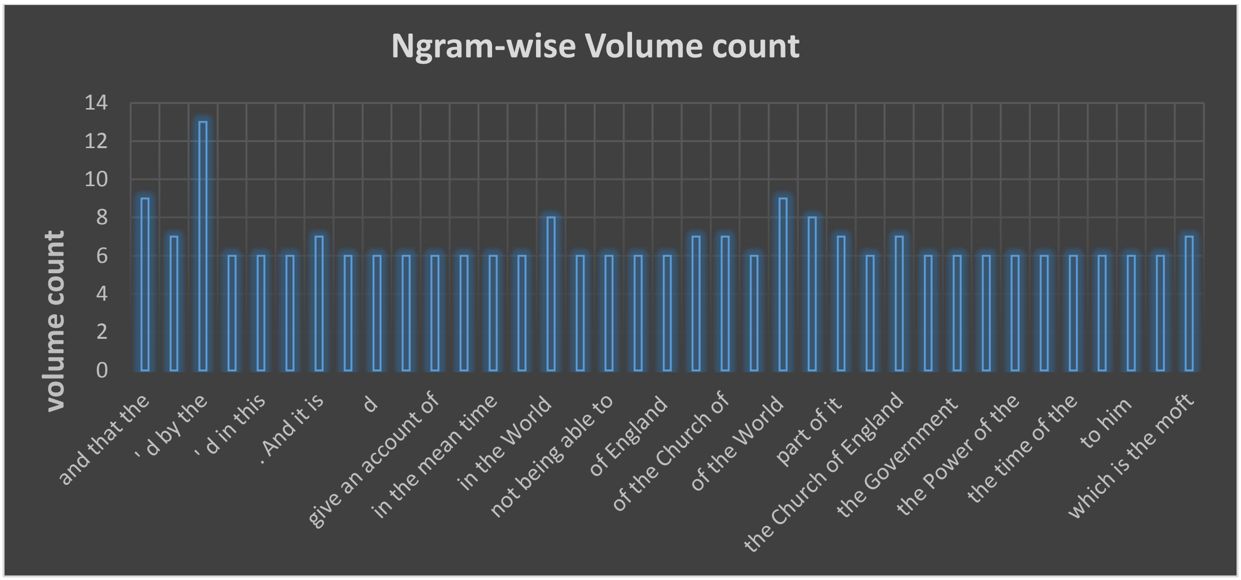
select ngram,MAX(volume_count) as max_volume,MIN(volume_count) as min_volume from ngrams where year between 1520 and 1700 group by ngram > /home/rakas1s/project/batch_views/batch_view2.csv

Piping files to provide it to Serving layer for further processing:

```
cat project/batch_views/batch_view2.csv | sed -e 's/\t/,/g' > project/batch_views/batchview_a.csv

cat project/batch_views/batchview_a.csv | sed -e 's/"//g' | sed -e 's/,,/,/g' >
project/batch_views/batchview_b.csv

rm -f project/batch_views/batchview_a.csv

cat project/batch_views/batchview_b.csv | awk -F ',' '{printf("%s,%i,%i;\n",$1,$2,$3)}'\ >
project/batch_views/batchview_c.csv

rm -f project/batch_views/batchview_b.csv

mv project/batch_views/batchview_c.csv project/batch_views/batch_view2.csv

rm batch_view2.csv batch_view2.sql
```

all these above shell commands are placed in script *batch_view.sh* under path */home/rakas1s/project/batch_views/*

The graph below describes the Ngram_wise Volume count. Which depicts the ngram that occurred in respective number of volumes in between the years 1520 and 1700



Serving Layer:

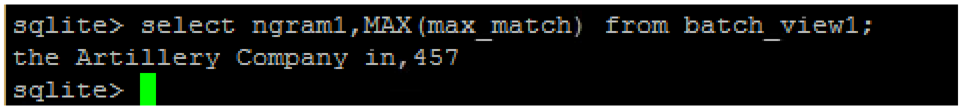
Placing the generated batch_views on to the serving layer (SQLite)

```
sqlite3 bv.db < sqlite.sql
```

execute the above command in the following path ***/home/rakas1s/***

Contents of the file sqlite.sql is as follows:

```
DROP TABLE IF EXISTS batch_view1;

CREATE TABLE batch_view1(ngram1 STRING,max_match INT,min_match  mv INT);

.separator ","

.import  project/batch_views/batch_view1.sql batch_view1

select ngram1,MAX(max_match) from batch_view1;
```



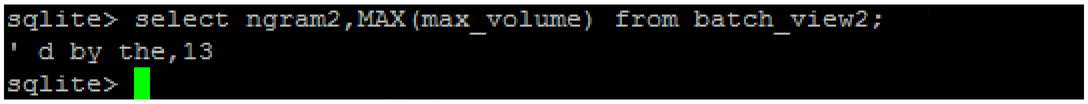
```
sqlite> select ngram1,MAX(max_match) from batch_view1;
the Artillery Company in,457
sqlite>
```

CREATE TABLE batch_view2(ngram2 STRING,max_volume INT,min_volume INT);

.separator ","

.import project/batch_views/batch_view2.sql batch_view2

select ngram2,MAX(max_volume) from batch_view2;



```
sqlite> select ngram2,MAX(max_volume) from batch_view2;
' d by the,13
sqlite>
```

select ngram2,dif, max_volume, min_volume

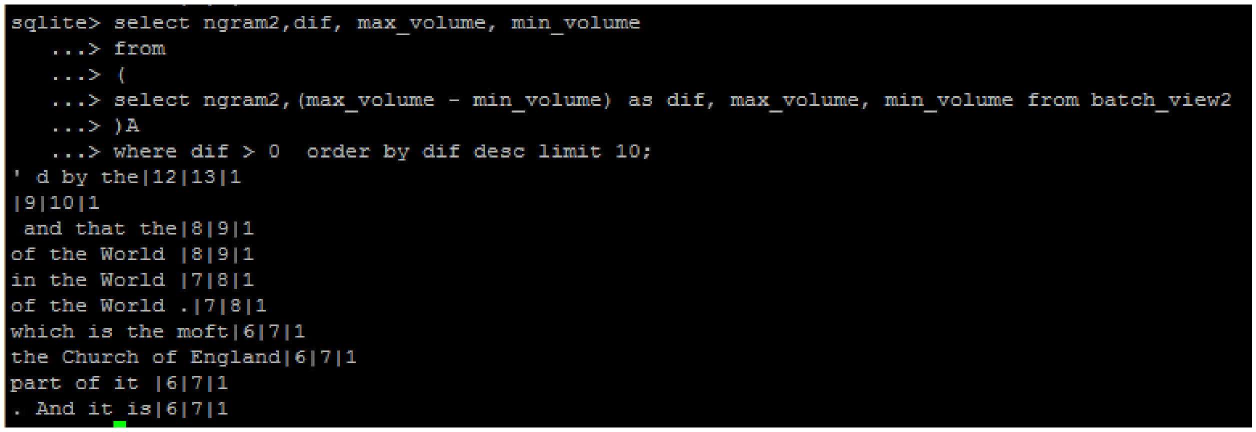
from

(

select ngram2,(max_volume - min_volume) as dif, max_volume, min_volume from batch_view2

)A

where dif > 0 order by dif desc limit 10;



```
sqlite> select ngram2,dif, max_volume, min_volume
   ...> from
   ...> (
   ...> select ngram2,(max_volume - min_volume) as dif, max_volume, min_volume from batch_view2
   ...> )A
   ...> where dif > 0  order by dif desc limit 10;
' d by the|12|13|1
|9|10|1
 and that the|8|9|1
of the World |8|9|1
in the World |7|8|1
of the World .|7|8|1
which is the moft|6|7|1
the Church of England|6|7|1
part of it |6|7|1
. And it is|6|7|1
```