Cloud Computing

Project Report

Mahmoud Mohammadi 800868389(mmoham12@uncc.edu)
Hossein Hematialam 800836301 (hhematia@uncc.edu)

Dr. Srinivas Akella

Fall 2015

List of Contents

1. Project Overview

The main idea of the project is predicting trend of page hits for Wikipedia pages based on page views statistics and links graph from simple Wikipedia. In addition, we will work on Twitter data to predict future trends variations. Furthermore, we will investigate that if there is any potential relation between Twitter trends and Wikipedia pages views.

To reach the project's goal the following steps done:

1. Data Gathering
2. Data Preprocessing and cleaning
3. Model Training and testing.

Because the main goal of the project is to predict some trends and variations in near future, we have to use a technique best fitted for prediction purposes. We primarily selected the regression method. There are some important factors affecting the quality of predictions using regression models. The first one is the variables used for prediction. We have chosen the Twitter keyword and Wikipedia page views per time slots (one hour in this project) as our main feature. The next factor is which regression model a linear or polynomial is appropriate for the project. Having the regression model, the next question would be how many time slots required to be considered in building the training model. To select the appropriate regression model we have to test and evaluate the results for each model and then comparing the results. In addition, to select the proper number of time slots required to have a good prediction of the following trends variation, some tests and comparisons should be accomplished.
In the model training step, all above mentioned test and comparisons have done and the results shown at the end of the report. Based on the final results we have chosen the linear regression technique with 3 number of previous slots required for predictions of top Wikipedia page views and 6 number of previous slots required for predictions of top twitter keywords.

2. Data Sets

At first we have to collect all data we need. As mentioned in our proposal we are planning to collect the Twitter and Wikipedia information.

2.1. Twitter

In the case of Twitter, we collected the tweets using the Twitter streaming API for one week. The Twitter. In order to get Twitter's data we need some account credentials and secrets, which can be obtained by creating an application via app.twitter.com. It is required to mention that using this API doesn't mean we can collect all the tweets world wide on that time, because Twitter only exports a very small portion (1%) of its total tweets via the streaming API. A Java application used to connect to streaming API. In this application we only collected the English tweets. The database system used to store these huge amounts of data is MongoDB. The output format of tweets is JSON and before storing the tweets in the database we parse the tweets and select only the text, created_at and hashtags fields and then store only the value of these fields.
So the primary data structure used for raw tweets is:

Extracted Fields from Twitter JSON data structure for each tweet:

Text	created_at	Hashtags
The contents of each tweet	Creation time of the tweets in receiver time zone	List of hashtags used in this tweet

2.2. WikiPedia

For Wikipedia, there is a storage containing number of page hits for each wiki page title which is organized in one hour time slots and can be downloaded from this address:

The positive point about this data is that the page hits number is accumulative number of al hits for a specific page title and so it simpler than the Twitter's data for processing. The structure of each line in these dump files is as:

Page Title	Page views	Page size
Main_Page	242332	4737756101

Wikipedia's page views dump files are organized in one hour compressed fies. In this case we only need the page title and page view hit from each line of that files.

2.3. Data Gathering Period

The data gathering period for Twitter is one week. From the 2015 November 27th to 2015 December 3rd.
During this period x MB of data collected for each hour and x GB for each day and totally x GB data for one week of data.

The data gathering period for Wikipedia is one week. From the 2015 November 27th to 2015 December 3rd.
During this period x MB of data collected for each hour and x GB for each day and totally x GB data for one week of data.

	Hourly	Daily	Total (one week)
Twitter	375 MB	9 GB	63 GB (10 million tweets)
Wikipedia	85 MB	2.04 GB	14.3 GB

3. DATA PREPROCESSING

Before using the collected data for training the prediction model and testing, it is required to clean and preprocess the data.

One step of data preprocessing done in the collecting phase by extracting the required fields out of many different fields. Therefore in this step we would handle simpler data structures than the primary raw data.
The mechanism and techniques used in a data preprocessing are based on what kind of data in which format we need for the training phase. In other words quality of cleaned data is determined by the properties of the training model.

The data structure and characteristics of them at the beginning of this step is as follows:

	Sorted	Primary Files	Additional Properties
Twitter	No sorting	Tweets in each Time Slot	-
Wiki	No Sorting	Page titles and view in each slot	-

3.1. Twitter Data Cleaning

The data were preprocessed and cleaned in the following ways:

- **Removing stop words**: Obviously many words in a tweet are pronouns, auxiliary verbs and adverbs that is required to be eliminated from the word count calculations. In addition some words used in conversions such as " wanna" ,"gonna" ,"you'll" and similar word are also should be eliminated as well. We evaluated some different stop word corpus such as ntlk package and some others but none of them could satisfy our need and so we use our own stop words word set by union of 3 different downloaded stop word including ntlk package and added our own words. This stop word set can be extended or modified for any purposes. Additionally, this stop words can be used as a white list instead of usual black list. And therefore for purposes that a specific cleaning based on a specific subject is required it can be applied.
- **Unicode Characters:** Although we collected the English tweets, there are some cases that some keywords not in the range of ASCII characters found in the tweets data. Currently all non-ASCII characters extracted from the words in each tweet
- **New Line:** New line character is basically not a problem for word counting but because we are going to parse the tweets in CSV files and some times the tweets are expanded in multiple lines, so it is required to extract this special character for proper word splitting and counting.
- **Hash tags:** We process the hash tag and its corresponding word as one keyword. It means that if the only difference of two words is the first character of "#", we assume there are one keyword and therefore eliminate the initial "#" character.

- **Usernames:** We consider the usernames in tweets, words starts with character "@, as keywords and not the user names.
- **Punctuations**: If there is only one punctuation character at the end of a word, we eliminate that character and count it as a word without that punctuation.

3.2. **Wikipedia Data Cleaning**: Because this fact that the Wikipedia page view files are almost have all data we need, the data cleaning for the Wikipedia files are much simpler. The only filter we applied to them is the Unicode characters extraction and only the ASCII characters in the page titles are considered. IN this way the page title will be changed but their corresponding page views count remains unchanged.

4. **Sorting and Selection**: After data get cleaned the next step would be applying some calculations and making the data ready for the following training and analysis steps. The calculations applied are as follow:

4.1. **Keyword counting**: The goal of this project is to predict the trends variations in very near future. Therefore, the data we need is the count of keywords and hash tags (as keywords). In other words, we are going to calculate the keyword counts for each tweet in one-hour time slots. In the case of Wikipedia, the page view hits are already calculated and so we don't need to modify and recalculate them.

4.2. **Time Slot identifiers** : The time slots for this project starts from midnight of Nov 27th and ends in Dec 3rd. This duration split into 7*24 one-hour slots and all the the calculations in the project is based on these time windows. For simplifying the processing we assigned number from 0 for hour 00 am in Nov 27th, 1 for hour 1am 27th and so on until number 167 for hour 23pm in Dec 3rd. In other words we have 168 time slots.

4.3. **Sorting:** A lot different keywords and their corresponding counts will be appeared in each slot but we have to limit the scope of our training and prediction model to a part of it with highest word count values. Actually, after all keywords counted for each one-hour slot, it is time to sort them in descending order so that selecting the important keywords becomes easier. Both od Twitter and Wikipedia cleaned data will be sorted in this step.

4.4. **Top 500 :** In the case of Twitter for each time slot the 500 keywords with highest counts will be selected. Also, for Wikipedia, the top 500 pages with highest page hits will be selected.

4.5. **Merging:** The last step in this phase of the project is merging all top 500 keyword and page titles, selected from the last step into two (Wiki and Twitter) files. Each file contains all keyword and page views sorted by at first time slot and then its corresponding count or hits value.

4.6. Sample Sorted and merged data

Data quality and structure at the end of this phase:

	Sorted	Result Files	Additional Properties
Twitter	Sorted based on word counts in each time slot	A merged file out of tweet s in all time slots in one week	- Word count for each keyword in each time slot - Normalized time slot identifier
Wiki	Sorted based on page view hits in each time slot	A merged file out wiki hits of all time slots in one week	Normalized time slot identifier

Tweet Data Structure:

Time slots identifier	**Keyword**	**Keyword count in one slot**

Wiki Data Structure:

Time slots identifier	**Page Title**	**Page view hits in one slot**

A sample of Twitter keyword file after sorting :

5 ending time slots of hour 23pm in Dec 2nd (slot 166) and 5 starting time(hour 00 am) slots of Dec 3rd (slot 167):

Slot	Keyword	Count
166	*using*	*103*
166	*santa*	*102*
166	*anymore*	*102*
166	*hurt*	*102*
166	*fashion*	*102*
166	*voting*	*102*
167	*mtvstars*	*3062*
167	*like*	*2453*
167	*love*	*2071*
167	*&amp*	*1990*
167	*lady*	*1860*
167	*gaga*	*1813*

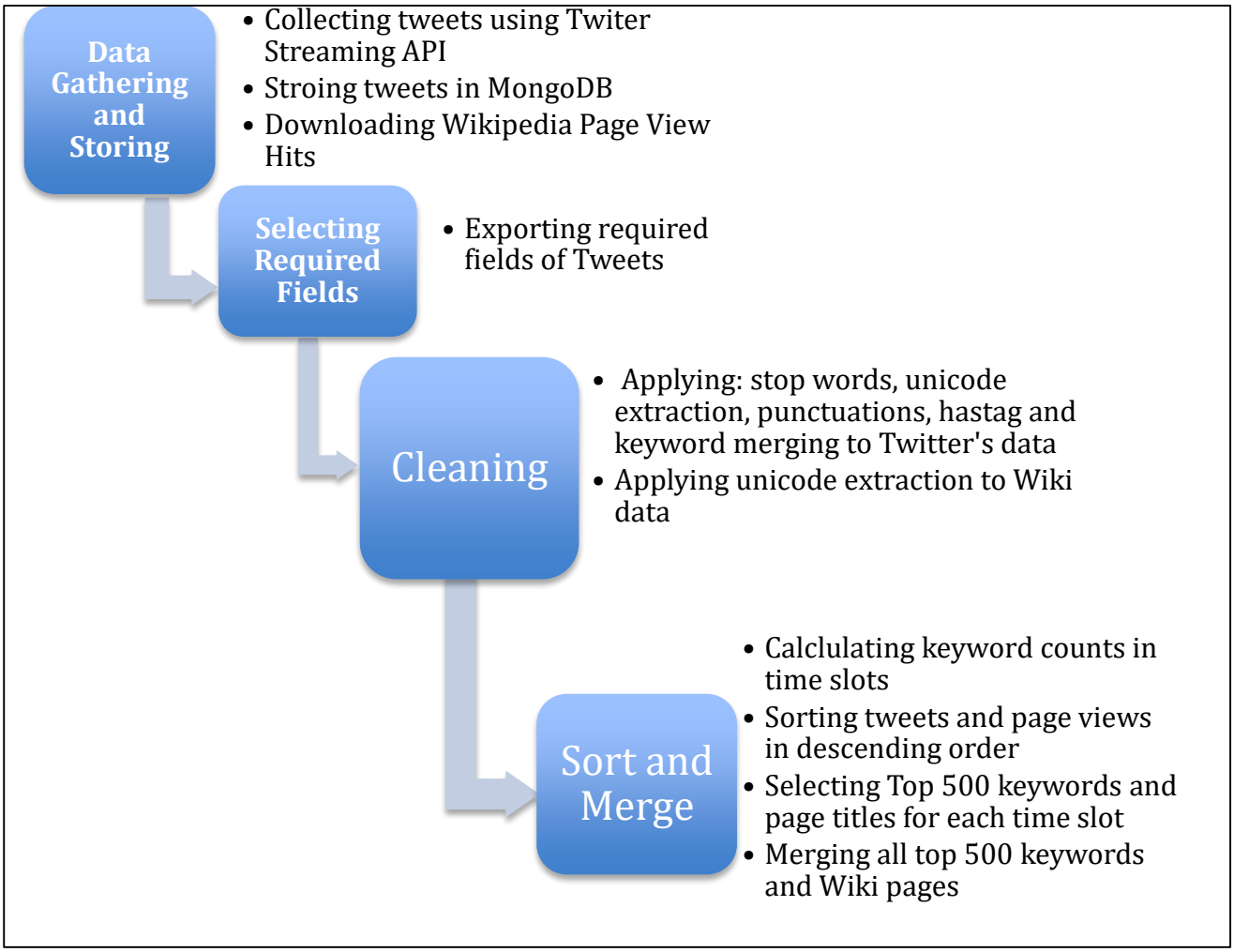
A sample of Wiki keyword file after sorting :

5 ending time slots of hour 23pm in Dec 2nd (slot 166) and 5 starting time(hour 00 am) slots of Dec 3rd (slot 167):

Slot	Keyword	Count
166	*Syria*	*313*
166	*Leonardo_da_Vinci*	*313*
166	*Edsel_Ford*	*313*
166	*Oprah_Winfrey*	*312*
166	*Unified_Task_Force*	*312*
167	*Befunge*	*5528*
167	*Java_(programming_language)*	*5475*
167	*index.html*	*5277*
167	*The_Game_Awards_2015*	*4187*
167	*The_Wiz*	*3769*

4.7. Flowchart of task done in this phase

This is the follow of all tasks done in the data gathering and preprocessing phases:



5. Model Training

5.1. Regression Model

5.1.1. Variables, Features

Step 1: In this project the features are the keyword counts per slot and page title hits per slot. In our model we created a set of word, its time slot, its count per slot and its previous counts per slot. For example, we had "45 mtvstars 6962" as an input line. Then we merged keyword time slots based like this:
"mtvstars, 45:=:6962#|###||##s44:=:4372 #|###||##s43:=:3745#|###||##s42:=:5473"

Step 2: We calculated the difference between time slots for each keyword and considered it as x in our regression model. For example, we got the output from previuos step, and created: "
mtvstars:=::=:45:=:6962#|###||##1:=:4372##||##2:=:3745##||##3:=:5473#||#""

Step 3: Then we split the output of previous step and extract a set of (x,y) pairs for each keyword in each slot. Then we applied linear regression and polynomial regression for them. By setting x=0, we calculated the predicted count for each keyword in each slot. We did these step by using n(2,9) previous count of keyword per slot.

Step 4: then we sorted the real value and predicted value to extract top 10 or 50.
For example we predicted top 10 keywords for Twitter in slot 45 with n=2 like this :

slot=45 real: [u'mtvstars', u'nicki', u'minaj', u'del', u'lana', u'rey', u'britney', u'spears', u'like', u'gaga'] predicted: [u'mtvstars', u'lady', u'gaga', u'&amp', u'like', u'nicki', u'minaj', u'love', u'del', u'lana']

Accuracy results for Wikipedia data set and twitter data set are as below:

	Linear	Polynomial	Linear	Polynomial
	Wiki top 10		Twitter top 10	
n=2	71.205		67.6506	
n=3	72.606	63.939	69.39394	58.36364
n=4	71.829	66.829	70.06098	63.90244
n=5	71.227	68.282	72.39264	65.58282
n=6	70.617	68.889	72.77778	67.09877
n=7	69.752	68.571	71.5528	69.00621
n=8	68.563	69	71.375	69.75
n=9	67.61	68.616	70.50314	69.24528
	Wiki top 50		Twitter top 50	
n=2	78.494		80.13253	
n=3	79.37	67.042	82.29091	70.66667
n=4	79.5	73.793	83.18293	76.45122
n=5	78.687	75.436	83.44785	78.67485
n=6	77.778	76.259	83.34568	79.75309
n=7	77.118	76.261	83.04348	80.86957
n=8	76.388	76.238	82.6625	81.125
n=9	75.836	75.874	82.06289	81.06918

Based on the accuracies, we conducted that linear regression works better for prediction rather than polynomial regression of order 2.
Additionally, with using 3-4 previous tie slots as training data, we get best accuracy to predict Wikipedia page views with linear regression. For Twitter data, optimal number of previous time slots is 6.

6. Project Product

Having the selected regression model , its corresponding variables and the collected data of previous time slots, we can predict the future trend variation at the desired time slot with accuracy percentages mentioned in the report. Based on the generated model, it means that at any given time slot, if we have the 6 previous time slots , we can predict the trend of both top 10 and 50 keywords of the next slot.

7. Tools and libraries

We have used different tools and libraries:

1. The Twitter4j package used to extract the tweets from the Twitter streaming API.
2. MongoDB used to store the tweets data(more than 63 G for 10 million tweets).To store the Wikipedia pages we have not used any DB systems.
3. In implementing the data cleaning, model selecting and its tuning variable we have used the Spark and Python.

8. Work Division

All the works done in a 50-50 divisions. Initially we discussed the steps , algorithms, data formats and all the steps we have to do and then each of us wrote our own part of the works

Hossein Hematialam	Cleaning Wiki Data, Implementing Regression Model and comparing the results
Mahmoud Mohammadi	Cleaning Twitter Data, Implementing Regression Model and comparing the results

9. Project Files and Descriptions

File Name	Description
CleanTweets.py	Reading tweet data exported from MongoDB and applying the Unicode extraction and primary file fomating
TweeterAnalysis.py	Reading from the previous prepared data, Applying the data cleaning logic (stop words, hashtag and keyword merging…) Sorting the tweets and selecting top 500 highest tweets per time slot
WikiAnlaysis	Reading Wiki files , applying the data preprocessing and cleaning logic, sorting the page title views, selecting the top 500 page titles for each time slot.
PageViewFile.java	Extracting English pages from page view zip file.
Linear.py	Get merged cleaned data as input. Create prediction value based on linear regression

	and create separate files for each time slot based on n(the number of data set used to predict the value)
Polynomial.py	It's similar to linear.py. The difference is it uses polynomial regression.
Sorting.py	It sorts the data and compare real value and predicted value for top keywords or pages in each time slot based on each n. The output is accuracy for each n.