Mining Tweets of Moroccan Users using the Framework Hadoop, NLP, K-means and Basemap

Abdeljalil EL ABDOULI, Larbi HASSOUNI, Houda ANOUN
RITM Laboratory, CED Engineering Sciences
Ecole Superieure de Technologie
Hassan II University of Casablanca, Morocco
elabdouli.abdeljalil@gmail.com, lhassouni@hotmail.com, houda.anoun@gmail.com

Abstract— **The information revolution and exactly the explosion of Web 2.0 platforms such as discussion forums, blogs, and social networks allow users to share ideas and opinions, express their feelings and much more. This revolution leads to an accumulation of an enormous amount of data that may contain a lot of valuable information. Much work has focused on analyzing these data, in particular those provided from social networks platforms like Twitter. In this paper, our objective is to propose an approach for analyzing the data generated by Moroccan users in the social network Twitter, in order to discover the subjects that interest Moroccan society and then locate on Moroccan map the areas from where come the tweets related to these topics. Analyzing the tweets of Moroccan users is a real challenge for two main reasons. Firstly, Moroccan users utilize for their communication in Twitter a variety of languages and dialects, such as Standard Arabic, Moroccan Arabic "Darija", Moroccan Amazigh dialect "Tamazight", French, Spanish, and English. Secondly, the Moroccan tweets contain a lot of URLs, #hashtags, spelling mistakes, reduced syntactic structures, and many abbreviations. In this paper, we propose an approach for detecting the relevant subjects related to Moroccan users by extracting the data automatically, and storing it in a distributed file system using HDFS (Hadoop Distributed File System) of Framework Apache Hadoop. Then we preprocess this raw data and analyze it by developing a distributed program using three tools, MapReduce of Framework Apache Hadoop, Python language, and Natural Language Processing (NLP) techniques. Afterward, we convert the corpus generated by the previous step into numeric features, and apply the k-means algorithm to cluster all words into general topics. Finally, we plot tweets on our Moroccan map by using the coordinates extracted from them, in order to have an idea about the geolocation of these subjects.**

Keywords: Framework Hadoop; HDFS; Distributed program; MapReduce; Python Language; Natural Language Processing; K-means.

I. INTRODUCTION

Twitter is a social network that has gained wide popularity in Arab world and especially Morocco due to its simplicity of use and services offered by its platform. For example, it allows publishing messages limited to 140 characters called "tweets" in which connected people can post links or share images.

According to the Arab Social Media Report [1], which provides statistics about the active Twitter users in the Arab World (22 Arab countries), the number of active Twitter users in the Arab world on March 2014 is estimated at 5,797,500 users. According to the same source, in Morocco, which is the country concerned by our research work in this article, the number of active Twitter users has reached 82,300 in March 2013. These statistics prove the activity of Moroccan users in the social network Twitter, which encouraged us to analyze the published messages.

In this paper, our proposed approach based on distributed system helps to get relevant information from posted messages on Twitter by Moroccan users. This information reflects the current situation and the main subjects that concern our Moroccan society. Our proposed system handles the streaming of the most recent tweets from Twitter platform using the open and accessible API of Twitter that returns well-structured tweets in JSON (JavaScript Object Notation) format, and stores all the tweets in our distributed system using HDFS [2]. The returned tweets are processed by a distributed program using MapReduce [3]. This program is written in Python language using Natural Language Processing (NLP) techniques [4], it's launched on MapReduce using the Pig UDF [5]. This processing is fundamental to clean these tweets of unwanted data and retrieve pertinent data; it also corrects the spelling mistakes and handles the linguistic diversity used by Moroccan users on Twitter. The result of the previous step is a clean corpus, the terms of which are then converted to numerical values in order to be clustered by the clustering algorithm K-means. Finally, we plot collected tweets on the Moroccan map using the coordinates extracted during the streaming step.

Our paper is organized as follows; we present some related work in Section II. In Section III, we introduce the tools and methods. In Section IV, we describe our distributed system. Finally, in Section V; we end with the conclusion and work in perspective.

II. RELATED WORK

Many researches have focused on the analysis of social network data in Twitter. [6] have found that, sometimes, users in the platform Twitter post news before the traditional media. [7] have introduced a method to calculate the frequency of terms presented daily in the corpus. Moreover, [8] have performed a semantic expansion of the terms presented in the tweets. Finally, [9] have proposed an approach to group the similar terms into one group; then identify those which are describing events.

978-1-5090-4062-9/17/$31.00 ©2017 IEEE

III. Tools And Methods

A. Apache Hadoop

Our system is based on the Apache Hadoop, which is an open-source software framework used for distributed storage (HDFS) and distributed processing (MapReduce) of massive data sets on computer clusters built from commodity hardware.

The HDFS (Hadoop Distributed File System) [2] system is highly fault-tolerant and designed using low-cost hardware. It's also designed to be available and scalable, and can store huge files reaching the terabytes. By default, each stored file is divided into blocks of 64 MB; each block is replicated by default in three copies. The HDFS is based on the architecture master-slave, and it consists of:

a) Single NameNode: running as a daemon on the master node, it holds the metadata of HDFS by mapping data blocks to data nodes, and it is the responsible of managing file system namespace operations.

b) Secondary NameNode: creates checkpoints of the file system present in the Namenode.

c) DataNodes: running as a daemon on slave nodes, they manage the storing of blocks within the node. They do all file system operations according to instructions received from the NameNode, and they send a Heartbeat and Block report on every file and block they store to the NameNode.

The MapReduce [3] is the heart of Hadoop. It's a program model for distributed computing based on Java, modeled after Google's paper on MapReduce. It's characterized by fault tolerance, simplicity of development, scalability, and automatic parallelization. It allows parallelizing the processing of large stored data by decomposing data submitted by the client into parallelized map and reduce workers. The input of the Map task is a set of data as a key-value pair, and the output is another set of data as a key-value pair. The input of the reduce task is the output from a map task. Between the reduce input and the map output, MapReduce performs two important operations, shuffle phase that covers the transformation of map outputs based on the output keys, and sort phase that covers the merge and sort of map outputs in reducers.

The MapReduce is also based on a master-slave architecture, and it consists of:

a) JobTracker: running as a daemon on the master node, its primary role is assigning tasks to TaskTrackers running on slave nodes where the data is stored. If the TaskTracker fails to execute the task, the JobTracker assigns the task to another TaskTracker where the data are replicated.

b) *TaskTracker*: running as a daemon on slave nodes, it accepts tasks (Map, Reduce, and Shuffle) from JobTracker. The TaskTrackers report the free slots within them to process data and also their status to the JobTracker by a heartbeat.

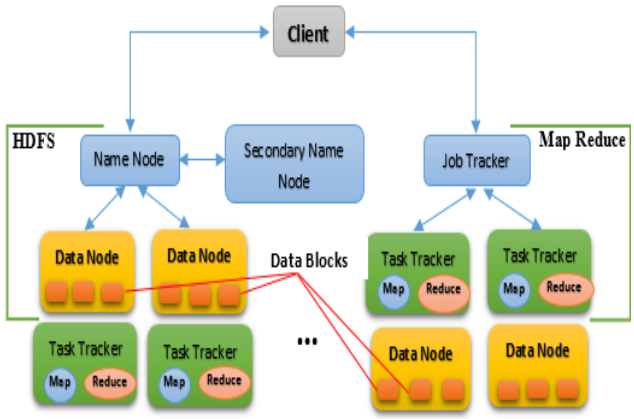


Figure 1. Apache Hadoop Architecture

Due to the complexity of the algorithms used and the quantities of data processed, the use of Apache Hadoop framework is necessary for the reliability of our system. It allows parallelizing the processing and getting better performance in less time.

B. Natural Language Processing (NLP)

Natural Language Processing [4] is a way for making computers recognize, understand, interpret and reproduce human language. NLP algorithms are based on machine learning algorithms and by utilizing them, developers can perform tasks such as topic segmentation, translation, automatic summarization, named entity recognition, sentiment analysis, speech recognition, and much more.

There are two components of NLP. The first component is Natural Language Understanding (NLU) that handles the mapping of given input in natural language into useful representations. The other is Natural Language Generation (NLG) that transforms a formal meaning representation into text that expresses that meaning. There are five steps in NLP [10]:

a) Lexical Analysis: involves identifying and analyzing the structure of words and dividing the whole text into paragraphs, sentences, and words.

b) Syntactic Analysis: analyzing and arranging words in a sentence in a structure that shows the relationship between the words.

c) *Semantic Analysis:* extracting the exact meaning or the dictionary meaning of sentences from the text.

d) Discourse Integration: handles the meaning of current sentence depending on the sentence just before it.

e) Pragmatic Analysis: analyzing and extracting the meaning of the text in the context.

Using the NLP in our system, we were able to process the tweets published by the Moroccan users in spite of the many difficulties they present and which we have mentioned previously. We processed the content of tweets returned in

JSON format from Twitter Streaming API after the extraction of pertinent information.

C. PIG UDF

Apache Pig [14] is a popular system for analyzing large data sets representing them as data flows by executing complex Hadoop map-reduce. It gives developers a high-level view by adding a layer of abstraction on top of Hadoop's map-reduce mechanisms, to perform all the data manipulation operations and simplifies complex tasks in Hadoop. Pig provides a high-level language known as Pig Latin for programmers who are not so good at Java. It is an SQL-like language which allows developers to perform MapReduce tasks easily and to develop their own functions for processing data.

A Pig UDF [5] (User Defined Functions) is a function that is accessible to Pig but written in a language that is not PigLatin like Python, Jython or other languages.

We use Pig UDF in our system to execute a complex processing program, written with Python language and based on NLP, in a distributed manner using the Hadoop MapReduce.

D. K-Mean

K-means [15] is a popular algorithm in document clustering. It is an unsupervised learning algorithm, where the user needs to decide a priori the parameter K that indicates the number of clusters. It involves grouping large sets of data into clusters of smaller sets of similar data. The K-means algorithm involves two steps as follow:

a) Step 1: Selecting centers by selecting k objects randomly, each becoming the center (mean) of an initial cluster.

b) Step 2: Clustering data by assigning each of the remaining objects to the cluster with the nearest distance. The most popular method for calculating is Euclidean distance [17]. Given two points p= (p_1, p_2, …, p_n) and q=(q_1, q_2, …, q_n), their Euclidean distance is defined as:

$$d = \sqrt{\sum_{i=1}^{n} (pi - qi)^2}$$

To cluster the result of NLP processing, which is a corpus that contains terms from all the tweets, into general topics, we vectorize it by converting all words into numeric features, in order to map the most frequent words to features indices and hence compute a word occurrence frequency matrix. This conversion is based on the implementation of TF-IDF [13] (term frequency–inverse document frequency).

E. Matplotlib Basemap

Matplotlib [16] is the most used Python package for 2D-graphics. It provides both a very quick way to visualize data from Python and publication-quality figures in many formats. It includes Basemap Toolkit, which is a library for plotting 2D data on maps that provides the facilities to transform coordinates to different map projections; then Matplotlib uses these transformed coordinates to plot contours, images, vectors, lines or points on the map.

IV. ARCHITECTURE OF THE SYSTEM

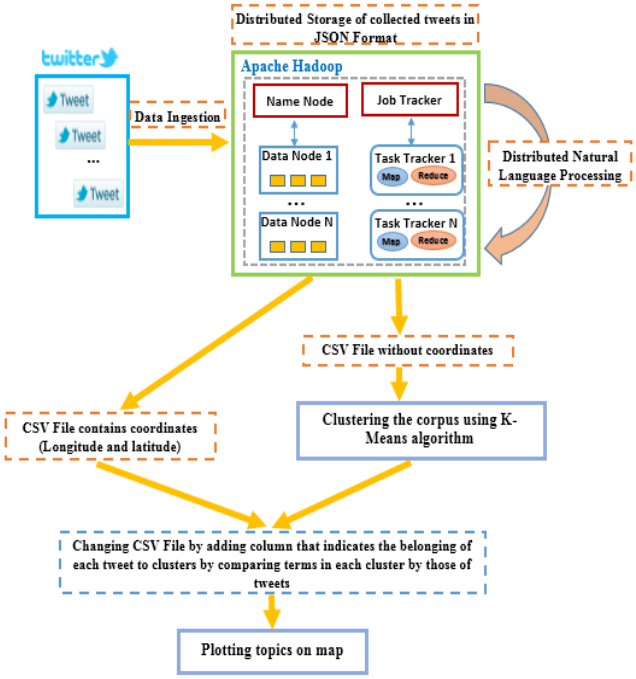


FIGURE 2. ARCHITECTURE OF THE SYSTEM

The first part of our system involves the extraction of data from the social network Twitter, which is the most important task in the data analysis process. All these tweets contain coordinates from different locations in Morocco and stored in the HDFS.

A. Extraction and distributed storage of raw data

1) Streaming Data from Twitter

To access the Twitter database, and stream tweets, we need to create an account on *https://apps.twitter.com*. For each created account, Twitter provides four secret information: consumer key, consumer secret key, access token and access token secret, then we are authorized to access the database and retrieve tweets using the streaming API.

To get the tweets we are interested in and which are those of the Moroccan users, we filter tweets by location using the Streaming API. We get the geographical coordinates (latitude and longitude) of Morocco that we utilized in this filter, by using the specialized website in geolocation *http://boundingbox.klokantech.com*. To handle the streaming of data from Twitter, we used Python library *Tweepy* [11] as shown in the script below, that allows accessing to Twitter API.

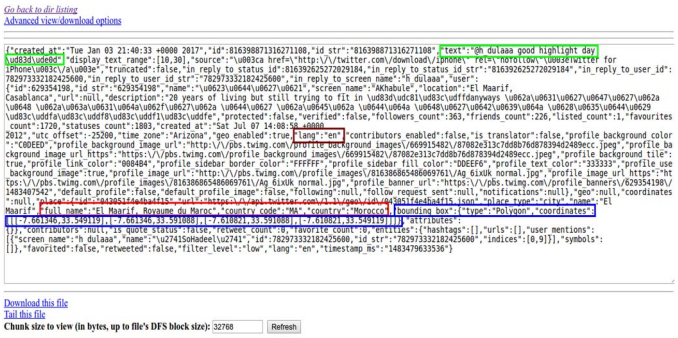
```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

auth.set_access_token(access_token,access_token_secret)
```

stream = tweepy.Stream(auth, l)

stream.filter(locations=[-17.2122302,21.3365321,-0.9984289,36.0027875],async='true',encoding='utf8')

During the streaming, by examining the content of tweets, we found that all the tweets contain the country code. We are therefore based on this code to extract the tweets coming from Morocco country whose code is "MA". Below, a sample of these tweets where we framed with different colors the informations useful for NLP processing.



2) Distributed storage using Apache Hadoop

The collected data from the platform Twitter are stored in a distributed infrastructure using the HDFS of Hadoop system.

The storage of tweets gathered from the Twitter platform in our distributed system is handled by using a Python wrapper for Hadoop called *Hadoopy* [12], which allows performing operations like reading and writing data to HDFS. The script below shows an example.

import hadoopy

hdfs_path = 'hdfs://master:54310/tweets/ '

hadoopy.put(Data ,hdfs_path)

B. Processing tweets with NLP

The second part concerns the processing of the stored data, starting with parsing the tweets and retrieving relevant information for our analysis, which are:

- *Text*: text of the tweet.
- *Lang*: language used by the user to write the tweet.
- *Coordinates:* location coordinates of a tweet.

The core of our system is the NLP; it contains all the necessary algorithms to handle the linguistic diversity that characterizes the communication of Moroccan users on social network Twitter.

The library used in our system to process tweets with NLP is the Natural language processing Toolkit (NLTK), which is a set of open-source Python modules, allowing programs to work with the human language data. It involves capabilities for tokenizing, parsing, and identifying named entities as well as many more features, it also provides over 50 corpora and lexical resources such as WordNet and a set of text processing libraries.

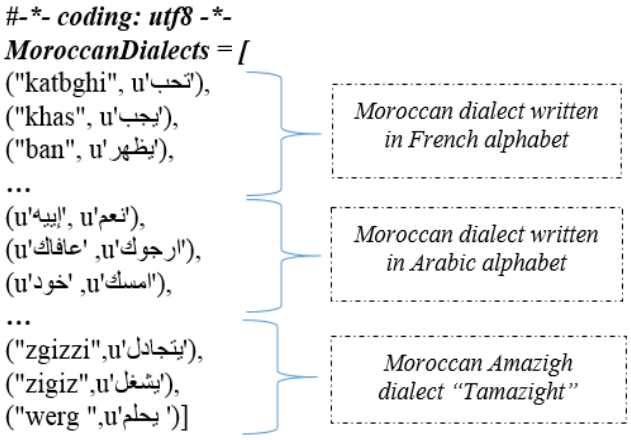
The steps we followed to process the stored Tweets are:

a) Delete unnecessary data: usernames, emails, hyperlinks, retweets, punctuation, possessives from a noun, duplicate characters, and special characters like smileys.

b) Normalize whitespace (convert multiple sequential whitespace chars into one whitespace character).

c) Convert hashtags into separate words, for example, the hashtag #MoroccanUsers is converted into two words Moroccan and Users.

d) Transform words written in Moroccan dialect, or in a dialect of Berber Tamazight into Standard Arabic. These words could be written using the Arabic or French alphabet. To perform this task, we create a python file that contains a dictionary of words that we gathered, then we store it in each slave node of our cluster and imported inside the NLP script executed in these nodes. Below, a part of this file.



e) Create a function to detect the language used to write the text of tweet (Standard Arab, French or English).

f) Create a function for automatic correction of spelling mistakes.

g) Create a list of contractions to normalize and expand words like *What's=>What is*

h) Delete the suffix of a word until we find the root. For example; *Stemming => stem*

i) Remove tokens of part of speech that are not important to our analyze by using the Part-Of-Speech software of Stanford University. This software reads the text and assigns parts of speech (noun, verb, adjective) to each word.

j) Remove stopwords for standard Arabic *(أن, ان, يعد...)*, French (*alors, à, ainsi, ...*), and English *(about, above, almost, ...)*.

These steps are assembled in a python file called NLTK_Tweet.py. This file is executed in a distributed manner by an Apache Pig file called Pig_Tweet.pig. The file NLTK_Tweet.py needs to be registered in the script of the Pig file using Streaming_python as follows:

```
REGISTER 'hdfs://master:54310/apps/NLTK_Tweet.py' USING
streaming_python AS nltk_udfs;
```

The launch of our file NLTK_tweet.py is defined as follows:

```
data = LOAD '/TwitterData/*  using TextLoader() AS
(line:chararray);

Result =  FOREACH data GENERATE
nltk_udfs.NLTK_Function(line));
```

C. Data clustering with K-means algorithm

1) Data

Twitter Streaming API allows to filter tweets by location or keywords and then will provide a live feed of tweets with the given keywords or specified location. Using this API, we collect experimentally a sample of 500 tweets based on the location filter. All collected tweets are stored in a distributed manner using HDFS.

2) Processing data with NLP

The second step is processing all stored tweets using our NLP program in a distributed manner using Hadoop's MapReduce implementation. A sample of NLP result is as follows:

	A	B	C
1	-8.002419	31.61341	مهاجر غير قانوني المغرب تسوية وضعية المرحلة الثانية
2	-9.677212	30.349652	GOUVERNEMENT BENKIRANE NEGOCIATIONS إنتخابات elections2016
3	-6.86848	34.002811	مشاورات تشكيل الحكومة إنتخابات المغرب
4	-4.991033	34.050369	Maroc politique régularisation migrants clandestins

3) Preparing Data for clustering with K-means

Before applying the clustering algorithm K-means, we need to create another CSV file that contains the processed data using NLP program without coordinates, which will be used as input to K-means algorithm. The code is as follows:

```
import hadoopy
import csv

hdfs_pathTweet=
"hdfs://master:54310/corpusTweetsCoordinates/part-m-00000"
local_pathTweet="/home /corpusTweetsCoordinates"
hadoopy.get(hdfs_pathTweet , local_pathTweet )

# Delete the first and second column (coordinates of tweet)
file_in = '/home/corpusTweetsCoordinates'
file_out = '/home/corpusTweetsWithoutCoordinates.csv'
with open(file_in, 'rb') as fin, open(file_out, 'wb') as fout:
   reader = csv.reader(fin)
   writer = csv.writer(fout)
   for row in reader:
      writer.writerow(row[2:]
```

An example of the output:

	A
1	مهاجر غير قانوني المغرب تسوية وضعية المرحلة الثانية
2	GOUVERNEMENT BENKIRANE NEGOCIATIONS إنتخابات elections2016
3	مشاورات تشكيل الحكومة إنتخابات المغرب
4	Maroc politique régularisation migrants clandestins

4) Finding the optimal number of clusters for k-means using silhouette index

There are different approaches to find an optimal 'K' value for K-means clustering, the famous and widely used by research community are elbow, silhouette and gap statistic methods. In this paper, we choose to use the silhouette index method [18] for finding the optimal number of clusters. The silhouette index method measures the quality of a clustering and provides a succinct graphical representation on how well each object lies within its cluster. Silhouette values range from -1 to 1 and the values near '+1' indicate a good clustering. The algorithm of this method is as follows:

```
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(NLTK_Result)

s = []
for n_clusters in range(2,10):
   kmeans = KMeans(n_clusters=n_clusters)
   kmeans.fit(X)
   labels = kmeans.labels_
   centroids = kmeans.cluster_centers_
   s.append(silhouette_score(X, labels,  metric='euclidean'))

plt.plot(s)
plt.ylabel("Silouette index average")
plt.xlabel("Number of clusters")
plt.title("Optimal number of clusters")
plt.show()
```

We apply this method on our processed data, and the result is as follows:

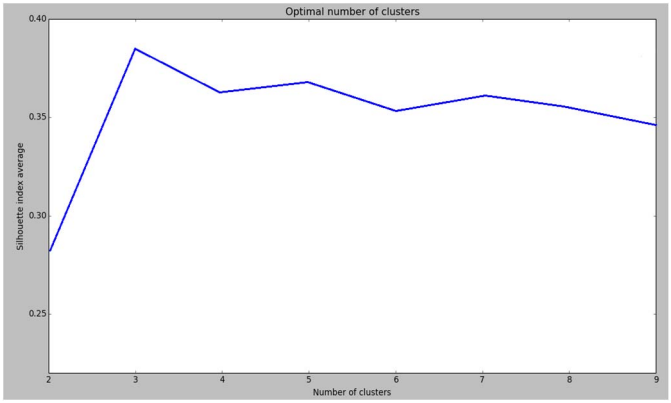


Figure 3. Graphical Representation of Silhouette index method

From this graphical representation, we can conclude that the optimal number of cluster is three because the silhouette index reaches the maximum value in k=3 comparing to other numbers of clusters.

5) K-means algorithm

k-means is one of the simplest algorithms to implement and to run which uses unsupervised learning method to solve known clustering issues. All we need is to find an optimal "k" and run it a number of times, then objects will be automatically assigns to clusters. It works really well with large datasets.

This step involves getting insights from all gathered tweets by vectorizing the previously generated corpus with TF-IDF (term frequency-inverse document frequency) [13] to measure the weight of each word and hence clustering the result with k-means algorithm based on the calculated tf-idf matrix.

To implement K-means algorithm in our system, we use *scikit-learn,* which is a machine learning library for Python that contains efficient tools for data mining and data analysis. First, we initialize K-means algorithm with the optimal number of clusters k=3 selected using the previous silhouette index method that indicates the optimal number of clusters. Each observation is assigned to a cluster to minimize the within-cluster Euclidian distance. The mean of observations is calculated and used as the new centroid. The calculation of centroid is repeated in an iterative process until the algorithm reaches convergence. The used algorithm is as follows:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(NLTK_Result)
true_k = 3
model = KMeans(n_clusters=true_k, init='k-means++',
max_iter=100, n_init=1)
model.fit(X)
print("Top terms per cluster:")
order_centroids = model.cluster_centers_.argsort()[:, ::-1]
terms = vectorizer.get_feature_names()

for i in range(true_k):
   print "Cluster %d:" % i,
   ClusterN = "Cluster %d" % i
   for ind in order_centroids[i, :10]:
      print ' %s' % terms[ind],
      fname_in = '/home/ corpusTweetsCoordinates'
      fname_out = ' /home/ corpusTweetsClusters.csv'
      with open(fname_in, 'rb') as fin, open(fname_out, 'a') as fout:
        reader = csv.reader(fin)
        writer = csv.writer(fout)
        for row in reader:
            for col in row:
              if terms[ind] in col.decode('utf8'):
                  writer.writerow([ClusterN]+ row)

   print
```

An example of the output:

Cluster 0: can, afrique, groupe, gabon, …
Cluster 1: formation, gouvernement , négociation, …
Cluster 2: politique, migrants, regularisation, ….

We use the clustering result to detect the belonging of each tweet to clusters by comparing words of clusters by those of tweets. Then we change the CSV file that contains tweets with coordinates by adding a column that indicates the cluster to which each tweet belongs. The CSV file used to plot the locations of these tweets is as follows:

	A	B	C	D
1	Cluster 0	-8.002419	31.61341	مهاجر غير قانوني المغرب تسوية وضعية المرحلة الثانية
2	Cluster 1	-9.677212	30.349652	GOUVERNEMENT BENKIRANE NEGOCIATIONS إنتخابات elections2016
3	Cluster 1	-6.86848	34.002811	مشاورات تشكيل الحكومة إنتخابات المغرب
4	Cluster 0	-4.991033	34.050369	Maroc politique régularisation migrants clandestins
5	Cluster 2	-7.479621	33.010231	can afrique groupe equipe maroc

D. Plotting clusters on map

During the streaming of tweets from the Twitter API, we extract the coordinates (longitude and latitude) of each tweet and store them in a separate CSV file. After clustering the corpus, we transform this file by adding a column to indicate the corresponding cluster of each tweet. We then use this CSV file by Basemap, to show locations of clusters on our Moroccan map. The tweets that belong to *cluster 0* are in red color and the others are respectively in green and yellow color. The developed program is as follows:

```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt

# empty lists for the latitudes and longitudes
latsCluster0, lonsCluster0 = [], []
latsCluster1, lonsCluster1 = [], []
latsCluster2, lonsCluster2 = [], []

llon = -17.2122302
ulon = -0.9984289
llat = 21.3365321
ulat = 36.0027875

map = Basemap(projection='merc', resolution='l',
area_thresh=1000.0,
          llcrnrlon=llon, llcrnrlat=llat,
          urcrnrlon=ulon, urcrnrlat=ulat)

map.drawcoastlines()
map.drawcountries()
map.fillcontinents(color='gainsboro',lake_color='aqua')
map.drawmapboundary(fill_color='steelblue')

x,y = map(lonsCluster0, latsCluster0)
map.plot(x,y, 'bo', markersize = 10,color='red')
x,y = map(lonsCluster1, latsCluster1)
map.plot(x,y, 'bo', markersize = 10,color='green')
x,y = map(lonsCluster2, latsCluster2)
map.plot(x,y, 'bo', markersize = 10,color='yellow')

plt.title("Location of Topics")
plt.show()
```

The Figure 4 below shows the result of plotting clusters on the Moroccan map :

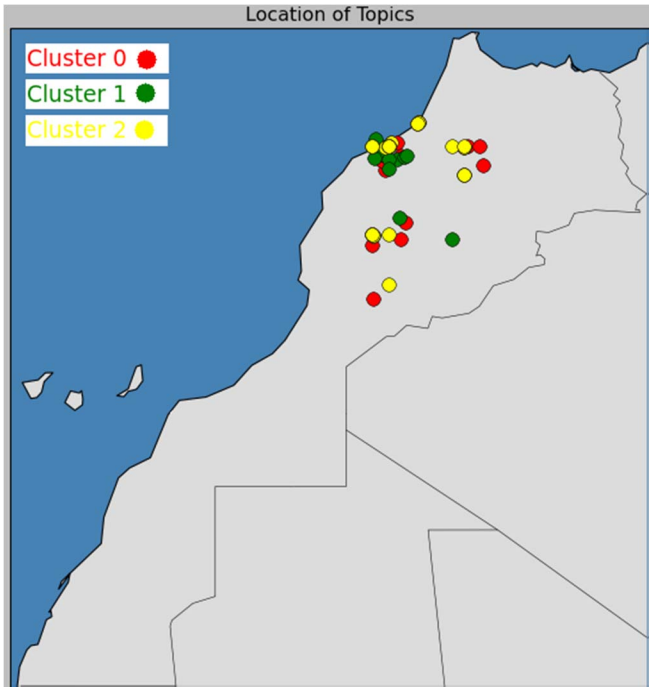


Figure 4. Locations of Topics on Moroccan map

This representation gives an idea about the hot topics that interest Moroccan people and their locations, which can led to better understanding of our Moroccan society.

V. Conclusion and Future Work

The purpose of our work is to develop a system that analyzes data extracted from the social network Twitter. This system allows to collect, process, and visualize topics that dominate in the Moroccan users' communication. It has to be noted that our system still needs improvement because of the growing usage of different languages and dialects. Moreover, Moroccan dialect is not stable which can lead to addition or change of meaning of some words. Therefore, enhancing the analysis of Moroccan dialect will require more future improvement.

VI. References

[1] arabsocialmediareport, "Twitter in Arab Region". [Online]. Available: http://www.arabsocialmediareport.com/Twitter/LineChart.aspx. [Accessed: 01- Jan- 2017].

[2] Mrudula Varade and Vimla Jethani, "Distributed Metadata Management Scheme in HDFS", International Journal of Scientific and Research Publications,Volume 3, Issue 5, May 2013.

[3] M. Ghazi and D. Gangodkar, "Hadoop, MapReduce and HDFS: A Developers Perspective", *Procedia Computer Science*, vol. 48, 2015.

[4] M. Nagao, "Natural Language Processing and Knowledge", *2005 International Conference on Natural Language Processing and Knowledge Engineering.*

[5] Pig.apache.org, "User Defined Functions". [Online]. Available: https://pig.apache.org/docs/r0.9.1/udf.html. [Accessed: 06- Jan- 2016].

[6] H. Kwak, C. Lee, H. Park and S. Moon, "What is Twitter, a social network or a news media?", *Proceedings of the 19th international conference on World wide web - WWW '10*, 2010.

[7] J. Weng and B.-S. Lee, "Event detection in twitter", *In 5th In. AAAI Conf. on Weblogs and Social Media,* 2011.

[8] O. Ozdikis, P. Senkul and H. Oguztuzun, "Semantic Expansion of Tweet Contents for Enhanced Event Detection in Twitter", *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2012.

[9] H. Becker, M. Naaman and L. Gravano, "Beyond trending topics: Real-world event identication on twitter", *In Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM11),*2011

[10] A. Chopra, A. Prashar and C. Sain, "Natural Language Processing", *International Journal of Technology Enhancements and Emerging Engineering Research*, vol. 1, 2013.

[11] "tweepy", github.com. [Online]. Available: https://github.com/tweepy/tweepy. [Accessed: 01- Jan- 2017].

[12] "hadoopy", hadoopy.readthedocs.org. [Online]. Available: https://hadoopy.readthedocs.org/en/latest/. [Accessed: 01- Jan- 2017].

[13] Z. Yun-tao, G. Ling and W. Yong-cheng, "An improved TF-IDF approach for text classification", *Journal of Zhejiang University Science*, 2005.

[14] pig.apache.org, "Welcome To Apache Pig". [Online]. Available: https://pig.apache.org/. [Accessed: 01- Jan- 2017].

[15] K. Žalik, "An efficient k′-means clustering algorithm", Pattern Recognition Letters, vol. 29, no. 9, pp. 1385-1391, 2008.

[16] matplotlib.org/basemap/, " Welcome to the Matplotlib Basemap Toolkit documentation". [Online]. Available: http://matplotlib.org/basemap/. [Accessed: 01- Jan- 2017].

[17] en.wikipedia.org/wiki/Euclidean_distance, "Euclidean distance". [Online]. Available: https://en.wikipedia.org/wiki/Euclidean_distance. [Accessed: 01- Jan- 2017].

[18] "Selecting the number of clusters with silhouette analysis on KMeans clustering — scikit-learn 0.18.1 documentation", Scikit-learn.org, 2017. [Online]. Available: http://scikit learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis. html. [Accessed: 01- Mar- 2017]