Developing a Framework for Analyzing Heterogeneous Data from Social Networks

Aritra Paul[1], Mohammad Shamsul Arefin[2], Rezaul Karim[3]
Department of Computer Science & Engineering
Chittagong University of Engineering & Technology
Chittagong 4349, Bangladesh

***Abstract*—Due to the rapid growth of internet technologies, at present online social networks have become a part of people's everyday life. People shares their thoughts, feelings, likings, disliking and many other issues at social networks by posting messages, videos, images and commenting on these. It is a great source of heterogeneous data. Heterogeneous data is a kind of unstructured data which comes in a variety of forms with an uncertain speed. In this paper, we develop a framework to collect and analyze a significant amount of heterogeneous data obtained from the social network to understand the behavioural patterns of the people at the social networks. In our framework, at first we crawl data from a well-known social network through Graph API that contains post, comments, images and videos. We compute keywords from the users' comments and posts and separate keywords as noun, verb, and adjective with the help of an XML based parts of speech tagger. We analyze images related to each user to find out how a user like to move. For this purpose, we count the number of users in an image using frontal face detection classifier. We also analyze video files of the users to find the categories of videos. For this purpose, we divide each video into frames and measure the RGB properties, speed, duration, frame's height and width. Finally, for each user we combine information from text, images and videos and based on the combined information we develop the profile of the user. Then, we generate recommendations for each user based on activities of the user and cosine similarity between users. We perform several experiments to show the effectiveness of our developed system. From the experimental evaluation, we can say that our framework can generate results up to a satisfactory level.**

Keywords*—Heterogeneous data; recommendation systems; cosine similarity; video categorization

I. Introduction

The term heterogeneous data comes from the concept of 'Big Data', is a term for data-set that are so large and complex that traditional data processing applications are inadequate to deal with them. It often refers simply in user's behavior analytics, predictive analytics or certain another data analytics method that extracts value from data. The concept of big data stands on 4V theory. This 4V are actually Value, Velocity, Variety, and Variability. Among 4V, heterogeneous data is directly related to Variety. It means a huge amount of data in different nature or formats. Most of the cases the data-set is unstructured including text, images, videos, audio files, web-links etc. And in this era of data science, one of the biggest sources of this heterogeneous data is Social Networks.

Nowadays online social networks are part of people's everyday life. It is a theoretical construct useful in social science to study relationships between individuals, groups, organization or even entire societies. The term is used to describe a social structure determined by such interaction. It is a medium where individuals interact with several groups of individuals by posting status, uploading images, videos, sharing feelings and so on. Interaction pattern varies from individuals to individuals. We can observe different interaction pattern for different groups of users. That means social networks are a great source of heterogeneous data. The importance of developing framework let us know the real interest of the users. Categorization of the user is very important for knowing the type of users, determine the type of multimedia they are most interested, how they interact with other users and by the place they visited we can get an idea of their interests. The methodology we use for the analysis can be applied not only to online social networking systems but also to any social multimedia sites like content sharing sites.

The analysis is done after crawling a huge set of data from a social network. The structure of the social networks emerged from their interactions shows how this knowledge can help to design new systems or improve the existing systems. However, we analyze user who interacts with each other through social networks. We study well to discover the user's behavior in online social networks.

Heterogeneous data never come in a structured way actually it's an unstructured data type. So handling this unstructured data is tough in a normal relational database. That's why we prefer Bigdata and NoSQL platform like Hadoop. Hadoop allows handling unstructured data with billions of rows and millions of columns. The motivation of our work is to understand user's behavior in online social networks and how it is changing over time. The research is carried out to achieve following goals:

- To develop a framework for analyzing heterogeneous data from social networks.
- To categorize the users based on their activities.
- To develop a short profile of the user.
- To develop a recommendation system for users.

In this paper in Section 2, we present related works in data analysis from social network, their advantages, and limitations. In Section 3, we enumerate our methodology concisely. Section 4 shows the experimental result of our efforts. In Section 5 we conclude the paper.

II. Related work

There are some categorizations and measurement analysis systems that are operating in a single language only. A. Mislove et al. [1] developed a system for the measurement and analysis of online Social Networks where they examine data gathered from four popular online social networks: Flicker, YouTube, LiveJournal, and Orkut. They tried to crawl the publicly accessible user links on each site and obtained a large portion of each social network's graph. They presented a large-scale measurement study including analysis of the structure of multiple online social networks. M. Maia et al. [2]studied well to discover user behavior in online social networks but the correlation of user's behavior within these networks is not considered widely. However, it has some limitation, 1st one is, it can crawl less amount of data and the 2nd one is, the analysis is based on only on the measure of friendship relation. Here authors developed a system to analyze the behavior of the YouTube user grouping them through K-based clustering algorithm. The advantage of this system is it can group user with similar behavior.

Benevenutoy et al. [3] designed a system which identifies influential users and their network impact. An interesting fact is knowing the influence of users and being able to predict it can be a strategic advantage for many applications. The most famous application to researchers and marketers is viral marketing. There is one main limitation of current research efforts on identifying influential users in social network analysis: lacking of an effective approach for modeling, predicting, and measuring the influence. Utilizing the session information, the author first examined the number of concurrent users, concurrent sessions, that accessed social networks. The beginning of each day is marked on the horizontal axis. They see a diurnal pattern with strong peaks around 3 PM, at all-times, there are at least 50 people who are using the social network aggregator service. At peak times, the number of concurrent users surpasses 700 which is more than a 10-fold increase over the minimum. Drops in usage on certain days indicate clear patterns like weekly patterns, where weekends showed a much lower usage than weekdays.The strong diurnal pattern in social network workloads has also been observed in accessing message and applications on the Social network and in the content generation of blog posts and answers in user-generated content websites. The system is designed with a probability of activity over time and it has little access to social network API. They worked on the passing time of the users. J. Thomas et al.[4] focus on the integrative analysis method for heterogeneous data. They explained two different methods one is for Bayesian network method and another one is multiple kernel-based method. The limitation is the author has not indicated any better methods but simply telling which one is harder to prove.

Author [5] developed a system to identify malicious posts from Facebook in real time. They have done this using Facebook graph API. It gives quite a good result but it has some limitation too. Their framework deals only with text posts considering there are no spam posts.

In [6] author offered a parallel computing based method to extract a social network individuals from fused data, by using cumulative association Data Graph. They implemented a supervised learning framework to parameterize the extraction algorithms. The advantage is data access methods are compared broadly.

In [7] author showed the limitations of finding patterns and comprehend the structure with many nodes and links. To overcome the limitations they identified, they offered a structured technique for structural analysis of social networks. The advantage is, the network layout is kept stable for each action so that, users can perceive patterns with a flexible interface.

S. A. Catanese et al. [8] author narrated the collection and analysis of huge data describing the connections between participants in online social networks. They approached two alternative methods which are well defined and also evaluated practically against the popular social network Facebook. For data crawling, they introduced two approaches one is BFS crawler and another one is Uniform crawler. However, it has some limitation and that is, they approached two methods for data crawling but they did not try parallel crawling which would make the output more relevant. Wilson et al.

[9] used the adoption behaviors referring to some activities or topics (tweets, products, Hash-tags, URLs, etc.) shared among users implicitly and explicitly such as users forwarding a message to their friends, sometimes recommending a product to others, joining some groups having similar interests, and posting messages about the same topics or issues, etc.

There are some drawbacks of existing social network influence models based on either static networks or the influence maximization diffusion process are most existing models are descriptive models rather than predictive models. There are very few models that are able to predict user's future influence. Using a discrete-time model to model diffusion process in continuous time is very computationally expensive. There are some system proposed and classified by the researchers. They classified system as:

- Crawling Social Network for Social Network Analysis Purposes
- Measurement and analysis of Social Network
- Characterizing User Behavior in online Social Networks
- Development of a social Network Crawler for Opinion Trend Monitoring and Analysis Purposes

A. Crawling from Social Networks

This crawling framework is proposed by A. Mislove et al. [1]. They introduced automated scripts and by using it on a cluster of 58 machines, they crawled the social graphs of some social network. They selected Flicker, LiveJournal, Orkut, YouTube for crawling data. They described their work through the collection and analysis of massive amount data, describing the connections between participants in online social networks. Alternative approaches to a social network data collection are defined and evaluated in practice against the popular Social network websites. They describe a set of tools that they developed to analyze specific properties of such social-network graphs, i.e. degree of distribution, centrally measures, scaling laws and distribution of friendship.

B. Social Network Analysis

Breadth-first-search (BFS) is a well-known graph traversal algorithm which is optimal and easy to be implemented, in particular for visiting unweight, undirected graphs. For these reasons, it has been adopted in several Online social networks (OSN) crawling tasks. Starting from a seed node, the algorithm discovers rest neighbors of the seed, putting them in a FIFO queue. Nodes in the queue are visited in order of appearance, so the coverage of the graph could be represented as an expanding wave front. This algorithm, concludes its execution when all the discovered nodes have been visited. For large graphs, like OSNs, this stopping condition would imply huge computational resources and time. In practice, we have established termination criteria which is a coverage of at least three sub-levels of friendships and a running time of 240 hours, so as resulting in an incomplete visit of the graph. Chau et al. [10] assert that an incomplete BFS sampling leads to a biased results, particularly towards high degree nodes. Even, our experimentation data acquired through BFS sampling does not show a statistically significant bias. We investigate this aspect in comparison with others and that is obtained using a sampling technique which is proved to be unbiased.

The Uniform sampling of a social network has been introduced by Gjoka [11], he provided proof of correctness of this approach but implementation details omitted here. Social network relies on a well-designed system of user-IDs assignment, spreading in the space of 32-bit range. So as the rejection sampling methodology is viable, this approach requires the generation of a queue of random user-IDs to be requested to a social network, querying for their existence. If so, the user and his/her friend list are extracted, otherwise, the user-ID is discarded and the polling proceeds to the next. The advantage of this approach relies on the independence of the distribution of user-IDs with respect to the distribution of friendship in the graph.

C. Characterizing user Behaviour in Online Social Networks

Here, we briefly provide a comprehensive view of users behavior in OSNs by characterizing the type, frequency, and sequence of activities users are engaged in. They [3] developed a new analysis strategy, which they call the clickstream model. This model is used to identify and describe representative user behaviors in Online Social Networks(OSN) based on clickstream data. The modeling of the system implies two steps. The first step is to identify dominant user activities in clickstreams. This step involves enumerating all features users engaged in on OSNs at the level of the basic unit, which they call user activity. They manually annotated each log entry of the clickstream data with the appropriate activity class (e.g, friend invitation, browsing photos), based on the information available in the HTTP header. Because a user can conduct a wide range of activities in a typical OSN site, they further tried to group semantically similar activities into a category by utilizing the web page structure of OSN sites (i.e, which set of activities can be conducted on a single page) and manually grouping related activities into categories. The next step of modeling is to compute the transition rates between the user activities. To represent the sequence in which activities are conducted, they built a first-order Markov chain of user activities and compute the probability transition between every pair of activity states. To gain a holistic view, they built a Markov chain which describes how users transition occurs from one category to another.

D. Social Network Crawler for Option Trend Monitoring and Analysis Purpose

This is a system prototype to analysis trend in a social network. The proposed system crawl data from Social network, indexes the data and provides a user interface where end users can search and see the trend of the topics of their choice. The main objective of this system is to propose a framework that can contribute to the improvement of the way government official and communicate in regard to service delivery in rural areas. The premise of this system is that if the government can keep track of the citizen's opinions and thoughts about service delivery, it can help improve the delivery of such services. This research and the implementation of the trend analysis tool is undertaken in the context of the Siyakhula Living Lab which is an Information and Communication Technologies for development intervention for Dwesa marginalized community located in the Eastern Cape province of South Africa.

III. Methodology

We have followed a modular approach for the development of this framework. The system architecture of this framework comprises 7 basic modules; Social Network Access Module, Data Crawling Module, Data Separation Module, Data Storing Module, Data Categorization Module, Developing profile Module and Recommending Module. The whole structure is shown in Fig. 1. We have used Hadoop HBase as the storage system and applied NoSQL which gives our system a dynamic property. We have initialized our storage system according to row key value and column according to our categorized data.

A. Social Networking Access Module

First of all, it is not trivial to tackle large-scale mining issues: for example, measure the crawling overhead in order to collect the whole Social network graph which is to be exact 44 Terabytes of data to be downloaded and handled. However, even when such data can be acquired and stored locally, it is non-trivial to devise and implement functions that traverse and visit the graph or evaluate simple metrics. For all these reasons, it is common to work with a small but representative sample of the graph. Extensive research has been conducted on sampling techniques for large graphs but, only in the last few years, some studies have a light on the partial bias that introduced standard methodologies.

First of all, we have to establish a network connection to get access from the social networks. The network connection will allow us to get access from the Graph API. Then from the Graph API, we need to download the RestFB package in order to get access from the Graph API. The Graph API gave the pathway to get access tokens for collecting data from the social network.

Then with the Java crawler, we have crawled data simultaneously from social network more than 15 days. The data then categorized into status, links, images, videos, and audios. We have used Hadoop database to store this data. Before storing this data in Hadoop we have organized reliable data into a

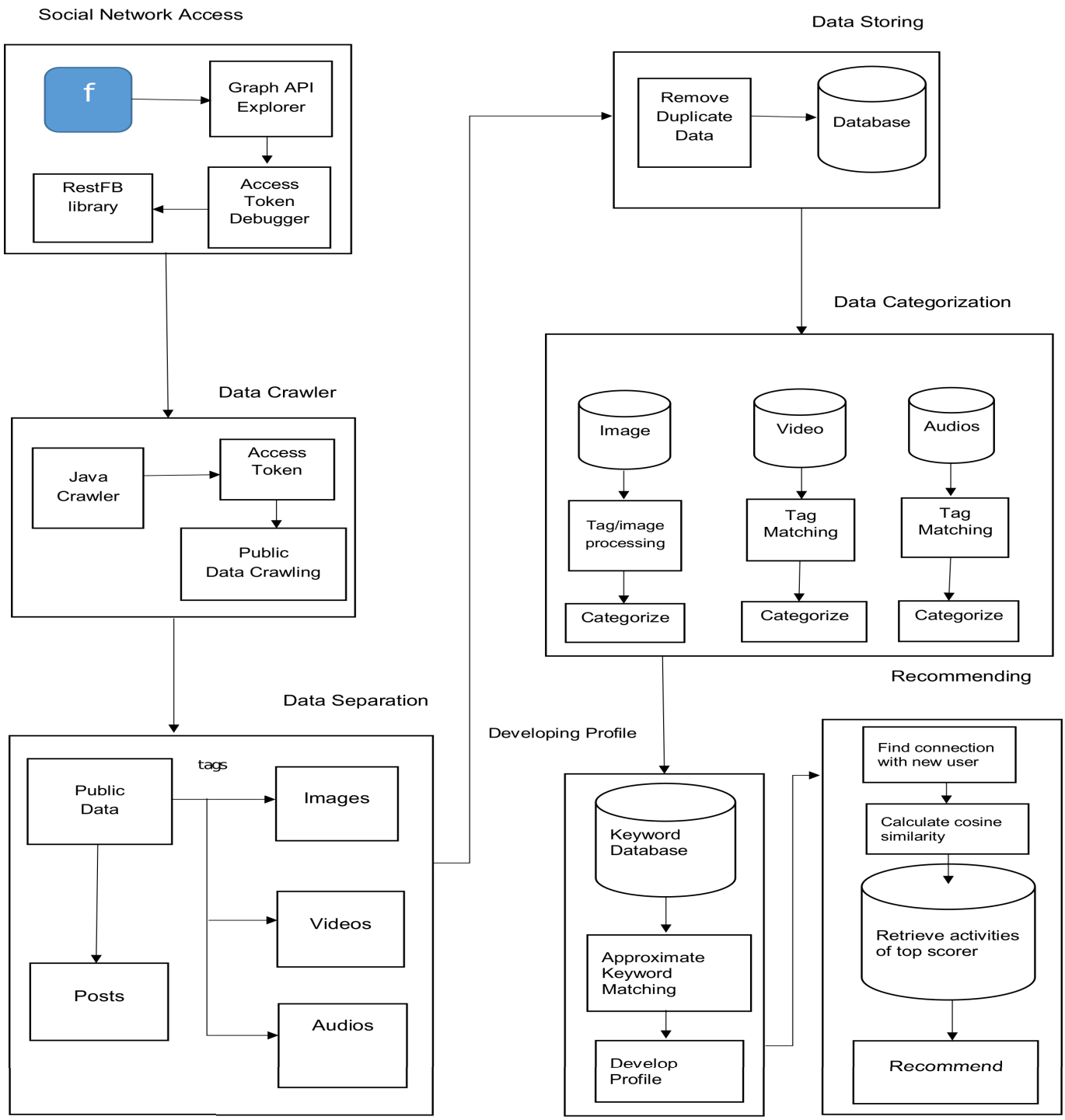

Fig. 1. System architecture

CSV file. Then load the CSV file into HBase by map reducing through bulk load method.

1) Data Crawling Module: Users can modify their privacy setting and Social network ensures several levels of information hiding; by default, users can only access private information of other users belonging to their friendship network, while friend list pages are publicly accessible. To maintain this status , Social network implements several rules, some behavioral terms which prohibits data mining, but the friend list can be dispatched through a script which asynchronously fills the Web page, preventing naive techniques of data extraction.

Companies providing social network services like Social network, build their fortune on behavioral and targeted advertising, exploiting the huge amount of information they own about users and their activity to show them normalized ads, to increase visibility and earnings of advertised

companies. This is why they are loath to share information about users and the usage of their network. Moreover, several questions related to user's privacy are abducted. The working algorithm is given below.

Algorithm 1 : Crawl Data
Input : User's access token
Require : Write data into a CSV file
Begin
call Graph API
call RestFB API
$accesstoken$ = "put token here";
$client \leftarrow accesstoken$
$user$ = "me";
$endpoint$ = "feed";
$u.csv \leftarrow$write$(user.get(data), endpoint)$;
End

Here we first collect user's access token through Graph API and store it in a variable. And after that we pull user's data through Rest API. Here we need an end point to crawl data and we use "feed". As we are crawling data through access token we keep user type as "me". We write this data in a CSV file simultaneously.

B. Data Storing Module

In Data storing module, there are two parts. One is keyword extraction another is storing. We extracted keyword from status by an XML based POS(parts of speech) tagger. We identified noun, verb, and adjective through this tagger. The keyword extraction algorithm is given below.

Algorithm 2 : Keyword Extraction
Input : Sentences and list of word
Require : Find the keyword
Begin
Load xml file;
Call SAX parser & parse the loaded xml file;
for each independent word **do**
SAXparser.parse (file, "word");
if true **then**
return(token);
else
continue;
end if
end for
remove("\\s", token);
write(token);
End

Here we extract keyword based on noun, verb, and adjective from a sentence. To extract this we use an XML based POS tagger named Word.xml. We parse a list of the word through SAX XML parser and if we found match we return the token from the XML file. Then we remove the symbols(<>) from the token and store it. Keywords are also stored in the database for future communication. We analyze video by converting video files into frames. Then we calculate each pixel to measure amount of red, green and blue color In the second part we stored our data separately in HBase after removing duplicate data if there exists any. It makes our data more reliable. Along with crawl data, this module also handles storage of important information for retrieval purpose. The data storage algorithm in HBase is given below.

Algorithm 3 : Store Data
Input : Crawled data (CSV file)
Require : Store the data
Begin
Create table through HBase shell name as Data, having field Name, Story type, Story, Status, ID, Time, Keyword, URL.
Exit from shell.
Copy CSV file from local to HBase.
Copy CSV file from HBase to Hadoop.
Call Bulk-Load method of Hadoop.
End

Here we first store our data in a CSV file to make our data more reliable. Then we store the file in Hadoop HBase by Bulk-load method. The bulk-load method uses the map-reduce algorithm which makes our data more reliable. There are columns like 'name', 'story type', 'story', 'status', 'id', 'time', 'keyword', 'url', etc. Under 'keyword'column there is 3 sub column: 'noun', 'verb', 'adjective'. So we first create our table with desired column family and column qualifier. Then we move our CSV file into Hadoop and call bulk load method to put all data into the table from the CSV file.

C. Data Separation Module

The data what we have crawled includes all type of data like images, videos, audios, posts, links both in English and Bangla; in a word everything. So before we store this data we have to separate this data and this separation is done based on tags. We also store this tags for the final data processing. The Data Separation algorithm is given below.

Algorithm 4 : Data Separation
Input : Crawled data
Require : Separation
Begin
Column family: status;
Column family: video;
Column family: image;
Column family: audio;
$i = 0$;
while $tag[i] \neq null$ **do**
if $tag[i] == \backslash status")$ **then**
$status \leftarrow content[i]$;
else
if $tag[i] == \backslash video")$ **then**
$video \leftarrow content[i]$;
else
if $tag[i] == \backslash image")$ **then**
$image \leftarrow content[i]$;
else
if $tag[i] == \backslash audio")$ **then**
$audio \leftarrow content[i]$;
end if
end if
end if
end if
$i \leftarrow i + 1$;

end while
End

Here we first create 4 column families: status,image, video, audio. Then we check tag type of a user's status and contents and according to the tags we keep keep status or contents in our desired column family.

D. Data Analysis Module

In Data Analysis module, we analyze categorized images, videos, based on tags and running further processing. There is two section. Image analysis and video analysis.

For image files We apply image processing to find out the number of people in images. We use OpenCV library and apply frontal face detection XML file(haarcascade) to count the number of people. We also apply OpenCV library to find out the length, duration, speed, total frame number, average height and width of frames etc. This process is beneficiary for recommendation module and to develop a specific short profile for the users. The algorithm is given below.

Algorithm 5 : Count People from Image
Input :User's image
Require : Number of people present in the images
Begin
$facecascade \leftarrow$ CascadeClassifier $('haarcascade.xml')$;
$imgcount = 0, total = 0$;
while $name == user$ and $url[i] \neq null$ **do**
$imgcount \leftarrow imgcount + 1$;
$img \leftarrow url[i]$;
Mat $src \leftarrow (file)img$;
$facedetection \leftarrow$face cascade.detect(src);
$face \leftarrow facedeection.length()$;
$total \leftarrow total + length$;
$i \leftarrow i + 1$;
end while
$avg \leftarrow total/imgcount$;
End

Fig. 2 shows an example of frontal face detection which helps us to determine the number of people present in the image uploaded by user. And based on its result we consider how a user likes to move. Does he likes to move with his friends most or not. and make an average for whole file. We also calculate frame speed, duration, height and width of the videos and for this we use some built in method of OpenCV(Cap_Prop_*). The algorithm is given below.

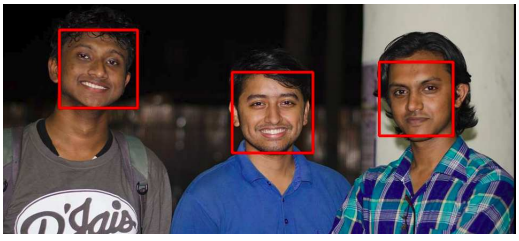

Fig. 2. An example of frontal face detection

TABLE I. AN EXAMPLE OF MEASURED VIDEO PROPERTIES

User	FPS	Frame no	H	W	R(%)	G(%)	B(%)
Rajib	17.9	677	420	420	38.2	32.5	29.3
Niloy	30	909	240	240	36.2	33.3	30.4
Fahim	30	909	240	240	36.2	33.3	30.4
Abir	25.05	474	352	352	36.2	32.4	31.4
Jawad	14.22	574	400	400	35.6	34.2	30.2

Algorithm 6 : Video Analysis
Input :User's video
Require : Properties
Begin
while $name == user$ and $video[i] \neq null$ **do**
$VideoCapturecapture \leftarrow path$;
while capture is open **do**
$fps \leftarrow capture.videoio, Cap\_Prop\_FPS$;
$number \leftarrow capture.videoio, Cap\_Prop\_Frame\_Count$;
$duration \leftarrow number/fps$;
$height \leftarrow capture.videoio, Cap\_Prop\_Frame\_Height$;
$height \leftarrow capture.videoio, Cap\_Prop\_Frame\_width$;
end while
initialize a 2D matrix;
$counter = 0, red = 0, green = 0, blue = 0$;
for $ilwidth$ **do**
for $jlheight$ **do**
$rgb[] \leftarrow$ getPixel$(capture, i, j)$
$red \leftarrow red + rgb[0]$;
$green \leftarrow green + rgb[1]$;
$blue \leftarrow blue + blue[2]$;
end for
end for
$total \leftarrow red + green + blue$;
$avgred \leftarrow red/total$;
$avggreen \leftarrow green/total$;
$avgblue \leftarrow blue/total$;
end while
End

Table I shows an example of measured video properties from user video files. We consider the average value here. Here 'H' means frame height, 'W' means frame width, "R" means red color, "G" means green color, "B" means blue color.

E. Profile Module

We have stored a keyword data-set and that data-set is used for matching. The keywords will be matched with the categorized data-set. When we will found more than one keyword in one user's data-set we will assume that user into a predefined category and that will help to develop a profile for the user and this task will be done under Develop Profile module. The working algorithm is given below.

Algorithm 7 : Profile Development
Input :User name and keyword list
Require : Select the desired data table
Begin
apply single column value filter for the user;
while $keyword_list \neq null$ **do**
apply single column value filter for the proper keyword;
initialize filter list;
$filter_list_property \leftarrow$ MUST PASS ALL;

$filterlist \leftarrow$ single column value filter;
scan $(filter_list) \leftarrow$ table
if $result \neq null$ **then**
write($category$);
else
go back to the beginning of while ;
end if
end while
End

Here we filter out the user and keyword through single column value filter which is a special filter in HBase. Then we add this filters into a filter list which must have to pass both filters and then scan it against our table. The profile of a user depends on the return value of the scanned result.

F. Recommendation Module

If a new user joins in a social network we want to recommend movies, travel places, music, or story books for him/her. In the recommend module, we do this two different way. Considering (i) status and comments (ii) image and videos. Here we first normalize user profile and convert that into vector plane. Then we calculate the score based on the cosine similarity between selected user and rest of the user simultaneously.

We first analyze user's profile based on status and posts and run Cosine Similarity to find out best match among other users. To find cosine similarity we first normalize users profile and convert it into vector plane. Then we check cosine similarity between user and other user and generate a score between 0 and 1. And based on this score we tried to show top five matched user in a table.

Secondly we consider Images and video uploaded in user's profile. We categorize videos into three types based on the tags: funny, musical and sporty. Then we process videos and find out its total frame number, length, height, width, speed and RGB properties and make average of them. Then we check cosine similarity again and based on top score we generate a second table and this time instead of status and image we consider only image and video. If we find same user in this two table then we recommend user the activities of those users. If no match is found then we suggest top users activity only. The algorithm is given below.

Algorithm 8 : Recommend User
Input :User name and keyword list
Require : Select the desired data table
Begin
if $selected_user \neq null$ **then**
$scan \leftarrow user$;
$vector \leftarrow (Vector)user_profile$;
$filterlist \leftarrow$ (table.$user \neq selected_user$);
$i \leftarrow 0$;
while $filterlist(i) \neq null$ **do**
$listA \leftarrow$ CosineSimilarity $(user, filterlist(i))$;
$listB \leftarrow$ Cosinesimilarity(file($user$), file($filterlist(i)$));
sort($listA$);
sort($listB$);
$listC \leftarrow$duplicate of $listAandlistB$;

TABLE II. MEASURING COSINE SIMILARITY BASED ON STATUS AND COMMENT

Name	Happy	Sad	Emotion	Tourist	Music	Movie	Food	Sporty	Score
Ratul	0	1	0	1	1	1	0	1	-
Fahim	1	1	0	1	1	1	0	0	0.79
Rajib	1	1	1	1	1	0	1	1	0.67
Kibria	0	0	0	0	1	1	1	1	0.67
Abrar	0	0	0	1	0	0	0	1	0.63
Riad	0	0	0	0	1	0	0	1	0.63

TABLE III. MEASURING COSINE SIMILARITY BASED ON IMAGE AND VIDEO

Name	Funny	Music	Sport	User1	User2	User3	....	UserN	Score
Ratul	1	1	0	1	0	1	0	0	-
Fahim	1	1	0	1	0	1	0	0	0.99
Shajal	1	0	0	1	0	0	0	0	0.82
Abir	1	1	1	1	0	0	0	0	0.82
Dip	0	1	0	1	0	0	0	0	0.71
Tarek	0	0	1	1	0	0	0	0	0.63

$i \leftarrow i + 1$;
end while
if $listC \neq empty$ **then**
showactivities($listC$);
else
showactivities($listA(0)$);
showactivities($listB(0)$);
end if
end if
End
function COSINESIMILARITY($(A[], B[])$)
for $i < A.length$ **do**
$dotproduct \leftarrow dotproduct + (A[i] * B[i])$;
$normA \leftarrow normA + (A[i] * A[i])$;
$normB \leftarrow normB + (B[i] * B[i])$;
end for
$result \leftarrow sqrt(normA) * sqrt(normB)$;
return $(dotproduct/result)$
end function

Here we first scan user profile and convert into vector plane. Then we filter out all users except recommending user. Then we run cosine similarity based on status comments and image, videos and keep the result in two different list. Based on score we find out same users from two list and keep them another list and suggest activities of this users to the recommending user.

Table II shows an example of cosine similarity measure procedure. Here we calculate cosine similarity of a user (Ratul) with respect to five other users. Here We consider only user's status and comments. And Table III shows cosine similarity based on user's images and videos.

G. Crawling and Information Retrieval in Bangla

Finding the relevant keywords from the Bangla text database has significantly different characteristics than the

same for English text database. In Bangla, there are so many variations of words. By just adding some postfixes, many different words can be formed. But these varied words have a similar meaning. So unless any mechanism is adopted, the varied words will be considered as distinct words. But as all the words have similar meaning, so all these words are relevant to a query with any of the words. Otherwise, queries with these different words will give different relevance value, which will degrade the retrieval efficiency. For this reason, it is very much important to find the root of the varied words having a similar meaning. These root words are used everywhere instead of all the varied words to keep necessary information for calculating relevance with the query. So it is absolutely necessary to find the roots of the words from the variations of the words by doing the morphological analysis for efficient information retrieval. But in English, there is a little variation of words. So it is not necessary to find the root of the words by doing the similar analysis.

On the other hand, there are many synonyms of words exists in Bangla. That means there are different words having different roots and the same meaning. So an efficient technique is required to manage the synonyms for efficient information retrieval. But if they are not managed in an efficient way then the system will treat all the synonyms of the same meaning as distinct words. This will degrade the efficiency of the information retrieval system.

Existing Bangla text database contains both Unicode and non-Unicode texts. It is difficult to search uniformly the database with both the type of text. But the required information may be found in any type of text. So it is very necessary to make a mechanism to handle both of the types of text uniformly.

H. Storage of Keyword Information into HBase

Keywords are used for different purposes. They are used to determine the data-set which are related to the analysis purposes. Keywords are also used to string matching. They are stored into the HBase by a column keyword. And this keyword column is sub divided into three sub-column: keyword: noun, keyword: verb, keyword: adjective. This three sub-columns are under keyword column family. Counting the frequency of the keyword in the data-set we find the strength of the keyword. The UserID, Public Post and No of String matched fields of the Data table represent UserID of public Post, Title, and number of String present in that data-set. In occurrence table, the field Keyword and No of the string matched is used to represent in how many posts a keyword occurs.

IV. Experimental Result

We crawled data directly from the social network with the help of access token provided by the user. We wrote it in a CSV (comma separated value) file first. Due to unstructured type of posts we found some bad lines. We also found some uncategorized character emerged from emoticon most probably. So we have to remove to run reliable analysis on the stored data. In this way we managed about 65 CSV files and after that, we merged all to a single CSV file by running command in the Linux terminal. It is also noticed that due to merging we lost some data. We tried several time but the result is same and the reason behind it is unknown. In Table IV, we have tried to show the correctness of data crawling of our system.

TABLE IV. Correctness (%) of data crawling

No of posts crawled	Bad Lines	Line removed for uncategorized character	Data lost for merging CSV files	Processed data found	Correctness (%)
10373	831	373	127	9042	87.17

TABLE V. Correctness of data analysis based on posts and comments

No	Language	No.of post crawled	No.of comments crawled	No. of posts and comments	Match found	Correct (%)
1	English	3645	448	39	3412	83.36
2	Bangla	4390	559	183	3121	63.06

We have considered posts and comments both in Bangla and English for the analysis. In this sense it is also a bilingual analysis framework. We used a bilingual dictionary for this task which has helped us to find out keywords. The more the words in dictionary the more it shows the accuracy. We used an XML file as dictionary. We stored about one lakh word with its property whether it is noun, verb, adjective, or adverb. However, we did not consider the all type of structure of sentences. In Table V, we showed the correctness of our data analysis based on posts and comments.

We have analyzed images and videos through OpenCV Java library. We tried to count the number of people presents in the images and videos and for that we have used Haar Classifier of OpenCV. It is a classifier which helps to detect frontal face of a person. We have counted the number of exists people through frontal face detection process. The result is not always accurate but reliable on average. Table VI shows the amount of experimental data as well as images and videos we have considered in this process and it's accuracy.

A. Performance of the Whole Framework

To calculate the performance of the whole system we have first calculated the performance of each module separately. Then combined all the result systematically. We have considered total 9042 posts and comments. The overall performance is showed in Fig. 3 and 4.

Where Fig. 3 shows the performance based on posts and comments, Fig. 4 shows based on person detected in images and videos.

V. Conclusion

The number of users in social networks has increased rapidly in last few years that makes social networks a great source of information. The data of these social networks are heterogeneous in nature as data in these social networks are

TABLE VI. Correctness of data analysis based on images and videos

No.	Data type	No. of file	Total person present	Detected person person	Correctness (%)
1	Image	325	912	698	76.54
2	Video	33	27	21	77.78

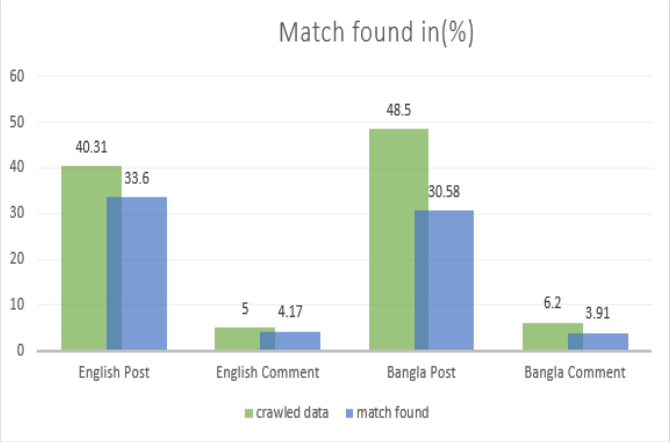

Fig. 3. Performance Measurement (Status & Comments)

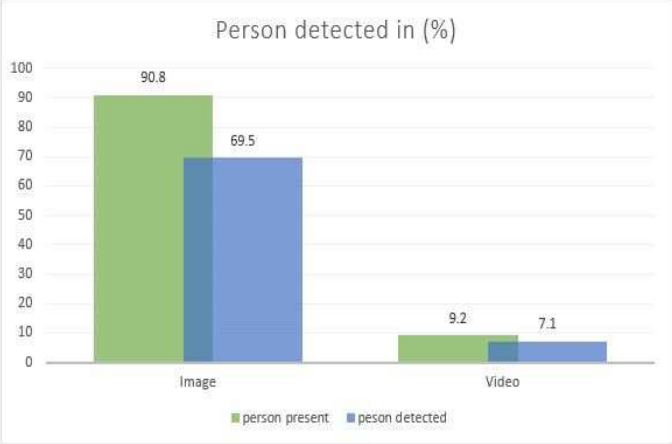

Fig. 4. Performance measurement (person detection)

of different forms such as text, videos, images etc. Proper analysis of these data can make the decision making tasks of the people and the organizations easier. Considering this fact, in this paper, we developed a heterogeneous data analysis framework to understand the users of social media, their interests, activities etc and based on these information of the users we generate recommendation for them. Our developed system can categorize the users based on their activities and can develop a concise profile of the users effectively. In future, we plan to analyze the performance of our system under large volume of heterogeneous data. In addition, we want to develop methods so that the system can work in multilingual domain.

REFERENCES

[1] A. Mislove, M. Marcon, K. Gummadi, P. Druschel and B. B. Bhattacharjee, Measurement and Analysis of Online Social Networks,IMC '07 Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, ACM, 2007, pp 29-42.

[2] M. Maia, J. Almeida and V. Almeida: Identifying User Behavior in Online Social Networks ,SocialNets '08 1st Workshop on Social Network Systems, 2008, pp: 1-6.

[3] F. Benevenutoy, T. Rodriguesy, M.Cha and V. Almeiday: Characterizing User Behavior in Online Social Networks,IMC '09 9th ACM SIGCOMM conference on Internet measurement, ACM 2009, pp 49-62.

[4] J. Thomas, L. Sael: Overview of Integrative Analysis Methods for Heterogeneous Data,2015 International Conference on Big Data and Smart Computing (BIGCOMP), 2015, pp: 266-270.

[5] P. Dewan, P. Kumaraguru,: Towards Automatic Real Time Identification of Malicious Posts on Facebook, 2015 13th Annual Conference on Privacy, Security and Trust (PST), 2015, pp: 85-92.

[6] A. Farasat, G. Gross and R. Nagi, Alexander G. Nikolaev: Social Network Analysis with Data Fusion, IEEE Transactions on Computational Social Systems, 2016, Vol 3(2), pp 88-99.

[7] A. Perer, B. Shneiderman: Balancing Systematic and Flexible Exploration of Social Networks,IEEE Transactions on Visualization and Computer Graphics, 2006, Vol 12(5), pp: 693-700.

[8] S. Catanese, P. De Meo, E. Ferrara, G. Fiumara, A. Provetti: Crawling Facebook for Social Network Analysis Purposes,WIMS '11 Proceedings of the International Conference on Web Intelligence, Mining and Semantics, 2011, Article no: 52.

[9] C. Wilson, B. Boe, A. Sala, K.P.N. Puttaswamy, B.Y. Zaho: User Interactions in Social Networks and their Implications,EuroSys '09 Proceedings of the 4th ACM European conference on Computer systems, ACM, 2009, Vol 2 pp: 205-218.

[10] D. Chau, S. Pandit, S. Wang and C. Faloutsos: Parallel Crawling for Online Social Networks, WWW '07 Proceedings of the 16th international conference on World Wide Web, 2007, pp: 1283-1284.

[11] M. Gjoka, M. Kurant, C. Butts, A. Markopoulou: Walking in Facebook:-A Case Study of Unbiased Sampling of OSNs, 2010 Proceedings IEEE INFOCOM, 2010, pp: 2498-2506.