Tweet Stream Summarization Based on Keyword Matching

Prashant S. Bagade\textsuperscript{a}, Prof. S. A. Shinde\textsuperscript{b}

\textsuperscript{a}Dept. of Computer Engineering SKNCOE, Savitribai Phule Pune University, Pune, India
\textsuperscript{b}Dept. of Computer Engineering SKNCOE, Savitribai Phule Pune University, Pune, India

Abstract

In recent years, number of users are being interested in the Social Networking site as well as micro blogging websites like Twitter, Facebook etc. In a single day Twitter counts Tweets over a million. The very complex part in this system is to control the real time applications sharing, keeping as well as managing large data. Because of huge amount of data generated by the user, it goes from different concentrating problems like noisy as well as frequent information. By the researchers view, Querying as well as retrieval of large information, it is very essential and one of the critical problems. Previous system tends to only work on the static as well as the limited information. Number of previous systems were tried to solve this problem and additionally given different solution over the problem. Summarization is a procedure consists of a text document in like a manner that short summary created using the essential keywords of the original document. To summarize information generated by the Twitter feeds, there is requirement of dynamic methodology. In this paper, we propose the new method that builds the appropriate content-based summary in limited period of time. In proposed system, specifically, we implemented multi-topic summarization over the online dataset and it needed minimum period of time by comparison with other previous systems. Our proposed system proves time efficient as compared with previous systems.

Keywords: Continuous summarization, tweet clustering, summary, Tweet stream, timeline.

I. INTRODUCTION

In social networking, twitter also is a highly used and an online social networking service that permits every user to send as well as read short 140-character messages. Social networks as well as microblogging services like Twitter, Facebook etc. generated huge quantity of short-text messages. Only Twitter receives millions tweets on a single day. There is no need to registration to read the Tweets, unregistered user can also able to read the Tweets. Registered users post their tweets and unregistered users also read them. Twitter users have accessibility via the website interface, SMS or mobile device applications. On the basis of analysis, Twitter is one of the ten most-visited websites in the month of May 2015 as well as Twitter have millions of users. The main benefit behind the Twitter is that tweets are implemented for data communication as well as sharing concepts and also individual’s point of view. Previous systems are designed to work over the static as well as limited information. Actually, the real time applications tasks such as sharing as well as managing large data are very complex task. This huge amount of information produced by the social networking sites are goes from the complex problems like noisy and irrelevant information.

On the famous topic, Twitter may generate millions of tweets and extends over weeks. So, a user has required to visit the millions of tweets that are not even related with the user and it is not probable each time. Thus, to prevent this problem a method given called as filtering. Even if filtering is permitted, searching for essential contents within huge amount of tweets is also very critical task and this has to face due to immense amount of unrelated tweets. We have one another probable solution called summarization for data overload issue. Summarization is a procedure consists with a text document in an order that short summary created through implementing the essential keywords over the original document. There is some examples of the search engines in which summarization methods used for example Twitter, Facebook and Google etc. Summarization has other categories such as document summarization, image collection summarization and video summarization. Here, we focused on document summarization. The major concept of the summarization is to search a characteristic and common subset of the information that presents unique data of the whole set. The preprocessing step is required to filter the datasets because the tweets are represented in cryptic and informal way. The stemming and stop word removal techniques are used for the preprocessing.

II. RELATED WORK

Tweet summarization method has two stages. First step needs tweet information clustering and second step performs summarization.

Various authors are also determines the algorithm for stream data clustering in literature. BIRCH is an algorithm that manages iterative lessening and clustering by implementing hierarchy’s algorithm. This algorithm is an unsupervised data mining algorithm [7].

These algorithms are specially implemented to execute continuous grouping on huge data sets. Implementation of BIRCH algorithm has a benefit that is, it has ability to create cluster in enhanced and fundamentally. It crates bunch of approaching and multi-dimensional metric data
A twitter post is as long as 140 characters in length and here we consider English posts. The twitter posts are casual, non-standard spelling and repeatedly do not have any accentuation. The hybrid TF-IDF based algorithm utilized for multi-post summaries of twitter post. Here few file abstract techniques are illustrated. Irregular Summarizer is a technique which self-assertively takes k posts or each subject for rundown. This framework was significant with a particular main objective to provide most cynical situation execution moreover set the lower bound of execution. Most recent Summarizer methodology takes the most recent k posts as summary from the determination pool. It may take the initial segment of a news article as summary. This system is executed in such a manner the brilliant summarizers can’t perform better than anything fundamental summarizer. This summarizer only utilizes the initial segment of the report as outline [9].

SumBasic methodology uses clear possibilities of word with an upgrade ability to process the best k posts. This approach is useful in case of it completely depends upon the repetition of words in the initial content. It is sensibly great fundamental. SumBasic framework was made by Nenkova and Vanderwende in 2005. This framework generates flat multi-record designs. Its design is impelled by the observation that, the words that are from time to time occur within the chronicle group with higher probability [1].

Real-time event summarization [4] provides data about event at whatever point any sub-events happen. This strategy is a two-stage procedure to reporting sub-events happen. Initial step is to distinguish sub-events as of late happen and in second step tweets about sub-occasions are chosen. Later consolidating these two stages we get summary of game from set of tweets.

LexRank summarizer uses a chart based framework. It recognizes pairwise equivalence between two sentences or between two posts. It creates the similarity score that is the largeness of the edge between the two sentences. The recent score of posts is designed in perspective of the weights of the edges that are connected with each other. This summarizer is helpful to give summary in perspective of graph instead of direct repeat layout. Regardless it relies on repeat; this framework uses the associations between sentences to incorporate more information. This is more difficult algorithm than recurrence based algorithm [2].

TextRank summarizer [3] is one another graph based strategy. This technique uses the PageRank algorithm. This provides another graph based summarizer that melds possibly a bigger number of information than LexRank. This is occurs in the way that it repeatedly modifies the weights of posts. The recent score of each post is liable to how it is recognized with immediately related posts and the way in which presents and are associated over various posts. TextRank joins the whole complexity of the graph rather than only pairwise similarities.

Twitter is a decent stage for individuals to express their point of view. The tweets are available in immense volume, it need push to comprehend what occur inside events. Here new strategies for summarizing events that discloses great correspondents and produces live sport upgrades from Twitter posts on events. Great correspondents chose logical tweets from dominant part of non-informative tweets [8].

Twitter streams additionally utilized for event summarization to present data in live way. The member based methodology is utilized for event summarization. The main segments utilized for summarization are member Detection, Sub-event Detection and Summary Tweet Extraction. Member detection distinguishes event members. Members are individuals take an interest in events. Sub-event detection recognizes sub-events relevant with members. The tweets are separated from sub-events utilizing Summary Tweet Extraction part [5].

ETS (Evolutionary Timeline Summarization) [10] is a web mining service that produces timelines for huge scale of information. ETS provides developmental directions over specific dates. ETS provides summery as indicated by score of timeline attributes. The benefit is that it encourages quick news browsing as well as information appreciation. ETS assignment has an adjusted optimization issue through iterative substitution.

Zhenhua Wang et al. introduce a summary framework called Sumblr. Sumbler is the predictable summary by stream clustering. This is the initial that determined constant tweet stream summarization. This framework involves three essential sections, to be particular the Tweet Stream Clustering module, the High-level Summarization module and the Timeline Generation module. Sumblr is profitable to work over dynamic, quick arriving and huge-scale tweet streams [11].

III. IMPLEMENTATION DETAILS

A. System Overview

Fig.1 demonstrates that proposed system contains three modules such as the tweet stream clustering module, the high-level summarization module and the timeline generation module. The tweet stream clustering module maintains the online statistical information. The topic-based tweet stream is given due to it has capability to effectively cluster the tweets as well as keep condensed cluster data. The high-level summarization separated within two kinds such as online and historical summaries. An online summary illustrates the trends between the public. So, the input for creating online summaries is
accessed directly from the ongoing clusters kept in memory and a historical summary useful for people to determine the major communication within a particular period of time, so we have to remove the effects of tweet contents from the outside of that period of time. Thus, to produce historical summaries is more difficult due to retrieval of the needed data for producing historical summaries.

B. Mathematical Model

Formulas:

Term Frequency tf(d) of term t in document d
The number of times that t occurs in d.

Inverse Document Frequency estimate the rarity of a term in the whole document collection

\[ idf_i = \log \frac{|D|}{|\{ j : t_i \in d_j \}|} \]

Where |D|= Total no: of documents j = no: of documents containing the term ti

Cosine Coefficient = \[ \frac{|X \cap Y|}{\sqrt{|X|^2|Y|^2}} \]

System S is represented as S= \{T, D, TSC, V, P, S, G\}

Process:
- **Input:**
  1. Input Tweet Stream T= \{t1, t2, t3, ..., tn\}

Where, T is the set of tweet streams and t1, t2, t3...tn are the number of streams.
- **2. Historical Tweeter datasets D= \{d1, d2, d3, ..., dn\}

Where, D is representing as a set of Tweeter datasets.
- **Process:**
  3. Tweet Stream Clustering TSC= \{V, P\}

Where, TSC is represent as a set of Tweet Stream Clustering and V= \{v1, v2, v3, ..., vn\} Where, V is represent as a set of tweet cluster vector and v1, v2, v3, ..., vn number of vectors and is the sum of weighted textual vectors. P= \{p1, p2, p3...pn\}

Where, P is representing as a set of pyramidal time frame and p1, p2, p3...pn number of frames. C= \{c1, c2...cn\}

Where, C is set of clusters generated using K-means algorithm.
- **4. High Level Summarization using TCV rank summarization algorithm.**

\[ S = \{O, H\} \]

Where, S is represent as a set of High Level Summarization, O= online summaries and H= Historical summaries
- **5. Timeline Generation using Topic detection evolution algorithm.**

\[ G = \text{Topic Evolution Generation.} \]

- **Output:** multi topic summarization

C. Algorithm

Algorithm 1: Multi-topic version of Sumblr

**Input:**Multiple topics (online or dataset),Number of cluster k;

**Output:** summary of Multiple Topic

**Process:**
1. While!topic.end () do
2. Topic t= topic.next ();
3. Study continuous tweet stream summarization.
4. For Tweet Stream Clustering module run Algorithm 2
5. Input the clusters CL generate using Algorithm 2 to Algorithm 3
6. ForHigh-level Summarization module run Algorithm 3
7. ForTimeline Generation modules run Algorithm 4.
8. Get the output of algorithm as Summary of multiple Topics.
9. END

Algorithm 2: Tweet stream clustering

**Input:** a cluster set C_set

1. While! stream.end () do
2. Tweet t= stream.next ();
3. Choose Cp in C_.set whose centroid is the closest to t;
4. If MaxSim (t) < MBS then
5. Create new Cluster Cnew = {t}
6. C_set.add (Cnew)
7. Else
8. updateCp with t
9. If TSeurrent%(ai) == 0 then
10. Store C-set into PTF.

Algorithm 3: TCV-Rank Summarization

Input: a cluster set D(c)
Output: a summary set S
1. S= Ø, T= All tweets
2. Build a similarity graph on T;
3. Compute LexRank scores LR;
4. Tc = tweets with the highest LR in each cluster;
5. While |S| < L do
6. Foreach tweet ti in Tc – S do
7. Calculateui;
8. Select tmax with the highest ui;
9. S.add(tmax);
10. While |S| < L do
11. for each tweet ti’ in T-S do
12. Calculate ui;
13. Select t 0 max with the highest u 0 i ;
14. S.add(t 0 max);
15. Return S;

Algorithm 4: Topic Evolution Detection

Input: tweet stream binned by time units
Output: timeline node set TN
1. TN =Ø;
2. While stream.end() do
3. Bin Ci = stream.next();
4. If hasLargeVariation() then
5. TN.add(i);
6. Return TN;

IV. RESULTS AND DISCUSSION

In this section we discuss our experimental results on different parameters. For our experimental analysis we extract online and historical tweets from twitter.

<table>
<thead>
<tr>
<th></th>
<th>Existing System</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (ms)</td>
<td>105657</td>
<td>54892</td>
</tr>
<tr>
<td>Memory (KB)</td>
<td>19739</td>
<td>9869</td>
</tr>
</tbody>
</table>

In above table comparison between existing system and proposed system on the basis of time and memory is shown. Proposed system is more efficient than the existing system.

Figure 2 graph demonstrates time comparison between existing system and proposed system. Previous system need more time to create single topic summary but in the similar amount of time proposed system creates multi-topic summaries.

Figure 3 graph demonstrates memory comparison graph. Memory graph demonstrates the memory comparison between the systems. Previous system needs more CPU consumption as compared with proposed system.

Figure 4 graph demonstrates time comparison graph. Previous system need more time to create single topic summary but in the similar amount of time proposed system creates multi-topic summaries.
V. CONCLUSION AND FUTURE SCOPE

Twitter has generated huge quantity of short-text messages. In real-world to control the real-time applications sharing, storing as well as managing huge amount of data is very complex task. Due to huge amount of data it goes from number of critical problems like noisy as well as frequent information. In this paper, we analyzed different methodologies for document summarization like filtering as well as tweet summarization. These methodologies are implemented to control the large amount of tweets. So, to avoid these problems, there is requirement to build a dynamic methodology to summarize information generated by Twitter feeds. In proposed system, multi-topic summarization over online dataset is that needs minimum period of time and memory as compare with previous systems. The proposed system is efficient than existing systems.

In future we try to improve our system to generate more meaningful summary and implement this framework on distributed system to evaluate large datasets.

REFERENCES