On Summarization and Timeline Generation for Evolutionary Tweet Streams

Gauri P. Gaurkhede
Department of Computer Engineering
Dr. D. Y. Patil Institute of Technology,
Pimpri,Pune-411018
Gaurigaurkhede45@gmail.com

Prashant G. Ahire
Asst. Prof., Department of Computer Engineering
Dr. D. Y. Patil Institute of Technology,
Pimpri, Pune-411018
ksprashantahire@gmail.com

***Abstract*— Short-text messages such as tweets are being created and shared at an unprecedented rate. Tweets, in their raw form, while being informative, can also be overwhelming. For both end-users and data analysts, it is a nightmare to plow through millions of tweets which contain enormous amount of noise and redundancy. In this paper, we propose a novel continuous summarization framework called Sumblr to alleviate the problem. In contrast to the traditional document summarization methods which focus on static and small-scale data set, Sumblr is designed to deal with dynamic, fast arriving, and large-scale tweet streams. Our proposed framework consists of three major components.**

First, we propose an online tweet stream clustering algorithm to cluster tweets and maintain distilled statistics in a data structure called tweet cluster vector (TCV). Second, we develop a TCV-Rank summarization technique for generating online summaries and historical summaries of arbitrary time durations. Third, we design an effective topic evolution detection method, which monitors summary-based/volume-based variations to produce timelines automatically from tweet streams. Our experiments on large-scale real tweets demonstrate the efficiency and effectiveness of our framework.

1. INTRODUCTION

With the fast popularization of Internet and great leap of network related technologies, Internet has changed people's lives worldwide, and Web 2.0 has changed the way we use Internet. Nowadays, people all round the world freely exchange information through Internet. In the developing history of Internet, text information has been playing an extremely significant role. Today it is still the most fundamental and main form of information in the Internet. Therefore, the demand of supervising, managing text information and using it as valuable resource has increased a lot rapidly – text stream analysis is now of great importance and practical value. Text stream analysis has several applications such as topic detection from a news stream, text crawling, document organization, topic detection & tracking (TDT), user characterized recommendation, user comments summary, trend analysis etc. The particularity of these analyzing tasks have in common is that text records come in the form of successive text sequence with time stamp. Result may be needed anytime as records everlastingly being generated. Clustering is one of the most important methods of data mining (Han & Kamber, 2006). Actually, the clustering problem has recently been studied in the context of numeric data streams and categorical data streams (Aggarwal, Han, Wang, & Yu, 2003, 2004; O'Callaghan, Meyerson, Motwani, Mishra, & Guha, 2002). Compared to traditional text clustering, in the text stream scene, challenges lie in several aspects: high algorithm efficiency is demanded in real-time; huge data set that cannot be kept in memory all at once; multiple scans from secondary storage is not desirable since it causes intolerable delays; and clustering algorithms need to be adaptive since data patterns change over time. The main contributions of this paper are as follows. First, analyzing of feature selection algorithm employed in the traditional text clustering shows that static features are not suitable for the text stream context in the long-time condition. Second, a text stream clustering algorithm TSC-AFS (text stream clustering based on adaptive feature selection) is proposed based on adaptive feature selection strategy extended from the traditional algorithm. Third, a text stream clustering system using TSC-AFS is present and proves effective with experiment.

The organization of the paper is as follows. In the next section, related works are reviewed and the limitation of using unchanging feature set in text stream clustering. In Section 3, based on adaptive feature selection, we present a text stream clustering algorithm TSC-AFS. In Section 4, we evaluate the performance of TSC-AFS, in experiment and analyze the results. In the last section, we conclude the paper and point out the future research.

A. Motivation

- Failure of traditional text summarization approaches in the context of tweets because of large volume of tweets as well as the fast and continuous nature of their arrival.

ISSN 2229-5518

- Need of a framework which will perform text processing on dynamic and large datasets.
- A lot of wastage of time for analyzing particular topic on social media like twitter
- No any sentiment model is available for giving review on topic using tweets as input.
- Need of a system which can generate summary on unstructured and redundant data by removing noise.

B. LITERATURE SURVEY

We have studied the paper “A framework for clustering evolving data stream”(C.C. Aggarwal, J. Han, J. Wang, and P. S. Yu) in which TCVs are considered as potential sub-topic; for stream clustering, Clustream method is used. It includes online and offline micro clustering component. For recalling historical micro cluster, pyramidal time frame also proposed for random time duration. [1]

In “BIRCH: An efficient data clustering method for very large databases” Clusters the data based on an in-memory structure called CF-tree instead of the original large data set. They proposed a scalable clustering framework which selectively stores important portions of the data, and compresses or discards other portions. [2]

Also we referred, “Text stream clustering based on adaptive feature selection” (L. Gong, J. Zeng, and S. Zhang) worked on a various services on the Web such as news filtering, text crawling, etc. It mainly focuses on topic detection and tracking (TDT).Clustering is used for analyzing text stream. [3]

In “A Probabilistic Model for Online Document Clustering with Application to Novelty Detection” in this paper we studied a probabilistic model for online document clustering. Nonparametric Dirichlet process prior to model the growing number of clusters, and use a prior of general English language model as the base distribution to handle the generation of novel clusters. [4]

For using function lexrank in TCV rank algorithm we have studied “LexRank: Graph based lexical centrality as salience in text summarization” (G. Erkan and D. R. Radev) in this paper lex ranking is calculated. Depending on the similar data graph is created. Lexrank is used for finding top ranked tweets among large data set. [5]

Also in “Document summarization based on data reconstruction” proposed to summarize documents from the perspective of data reconstruction, and select sentences that can best reconstruct the original documents. [6]

Lastly we have referred “on summarization and timeline generation for evolutionary tweet stream” we have referred Tweet Cluster Vector (TCV), TCV Rank algorithm, Topic evolution. In which TCV used for making effective clustering of tweet with the help of pyramidal time frame and tweet cluster vector, TCV rank summarization algorithm is used for generating online and historical summaries by evaluating top ranked function, depending upon top ranked tweets summarization is done. Topic evolution detection generates timeline by considering large variation of sub-topics in stream processing. [7]

C. PROPOSED METHODOLOGY

Developing non-stop tweet flow summarization is a hard mission to perform, due to the fact that countless number of tweets is vain, noisy as well as inappropriate in nature, because of the social manner of tweeting. Tweets are firmly related to their posted time and new tweets have a propensity to the touch base at a quick rate. Tweet streams are constantly extensive in scale, henceforth the summarization algorithm ought to be very proficient. Tweet streams are constantly significant in scale, henceforth the summarization set of rules must be very proficient. It have to give tweet summaries of subjective time spans. It ought to naturally recognize sub-topic changes and the minutes that they happen. In this paper we are going to develop a multi point variant of a constant tweet stream summarization system, mainly Sumbler to supply summaries and timelines of occasions with reference to streams, which will likewise reasonable in distributed frameworks and evaluate it on more finish and extensive scale data sets. The beyond variation of sumbler changed into no longer possible in disbursed range. The sumbler system which consist of three principle modules: the tweet stream clustering module, the highlevel summarization module and the timeline generation module. The tweet stream clustering module keeps up the online statistical data. The topic-based tweet stream is given; it is able to proficiently cluster the tweets and maintain up minimum cluster information. Two sorts of summaries are given by the high-level summarization module i,e online and historical summaries. An online rundown depicts what is as of now talked about among the general population. Hence, the input for creating online summaries is recovered straightforwardly from the present clusters kept up in memory. Then again, a historical summary helps people groups comprehend the principle happenings amid a particular period, which means we must dispense with the impact of tweet substance from the out of doors of that period. Therefore, restoration of the required facts for developing ancient summaries is greater confounded. The center of the timeline generation module is a subject evolution detection algorithm which provides real-time and variety timelines additionally.

A. *Architectur*

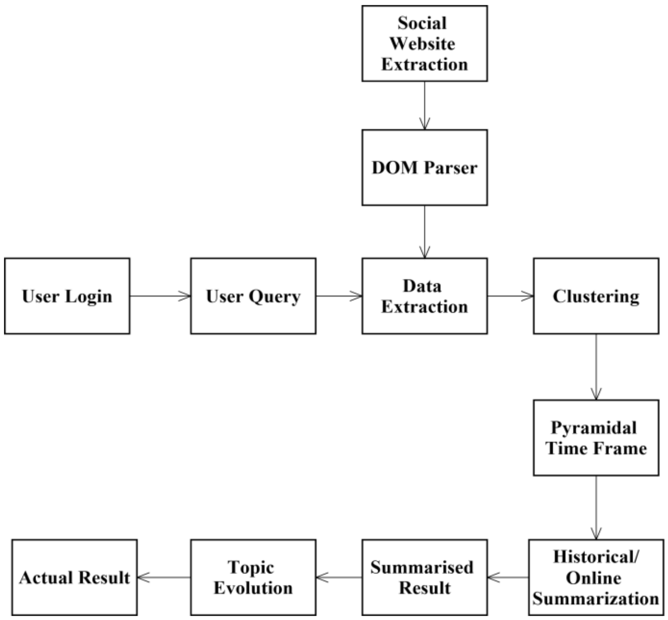

Fig. 1. Proposed System Architecture

B. *ProposedAlgorithm*

1. K-Means

K-means clustering is a type of unsupervised learning, which is used when you have unlabeled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively to assign each data point to one of *K* groups based on the features that are provided. Data points are clustered based on feature similarity. The results of the *K*-means clustering algorithm are:

1. The centroids of the *K* clusters, which can be used to label new data
2. Labels for the training data (each data point is assigned to a single cluster)

.

STEPS:

Step 1: Each document is represented as vector using the vector space model.

For example: TFIDF weight.

TFIDF: It stands for Terms Frequency Inverse Document Frequency, is a numerical statistics which reflects how important the word is to document in a collection or corpus.

a) Term Frequency: The number of times a term occurs in the document.

b) Inverse Document Frequency: Measure of whether a term is common or rare across all documents.

Step 2: Finding Similarity Score

Use Cosine similarity to identity similarity score of the Document.

Step 3: Preparing document cluster.

Step 4: initializing cluster center.

Step 5: Finding closest cluster center.

Step 6: Identifying the new position of cluster center.

2. Summarization algorithm:

The main idea of summarization is to find a subset of data which contains the "information" of the entire set. Such techniques are widely used in industry today. Search engines are an example; others include summarization of documents, image collections and videos. Document summarization tries to create a representative summary or abstract of the entire document, by finding the most informative sentences.

STEPS:

Step 1: The first step would be to concatenate all the text contained in the articles.

Step 2: Then split the text into individual sentences.

Step 3: Find vector representation (word embedding) for each and every sentence.

Step 4: Similarities between sentence vectors are then calculated and stored in a matrix.

Step 5: The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation.

Step 6: A certain number of top-ranked sentences form the final summary.

3. Sentiment Analysis:

Sentiment analysis is contextual mining of text which identifies and extracts subjective information in source material, and helping a business to understand the social sentiment of their brand, product or service while monitoring online conversations. However, analysis of social media streams is usually restricted to just basic sentiment analysis and count based metrics. This is akin to just scratching the

ISSN 2229-5518

Test Case ID	Test Case Name	Action Taken	Test Data	Expected Result
TC001	To test whether internet connection is available or not.	Enable internet connection.	Home Module	Internet connection should be available to access the application.
TC002	Enter empty value for Name	Name field is empty should be detected.	-	Display error message "Name required."
TC003	Enter empty value for Password	Empty password field should be detected.	-	Display error message "Password required."
TC004	Run the application.	Application should be run properly	Splash window displayed on the screen	Open Log in window
TC005	Open offline dataset file	File selection system window should be opened	Twitter.txt	Display tweets present in the dataset

surface and missing out on those high value insights that are waiting to be discovered

STEPS:

Step 1: Creating a Stream

a) Authenticate.

b) Build Stream.

Step 2: Data Cleaning

Tweets can contain many non-ASCII characters. Therefore, we need to sanitize it

Step 3: Sentimental Analysis

Library used: TweetInvi.

Step 4: Produce output (Positive, Negative, and Neutral).

4. Web Extraction Algorithm

Step 1: Understand what the real DOM of the web Page is.

Step 2: By using class name from DOM we can easily get all Comments from the code.

Step 3: Install HTML AgilityPack.

Step 4: By creating an instance of HTML web, load HTML file of give URL Using HTTP

Step 5: Loop through the tags and call the inner text property for each comment in the list

4. RESULT AND DISCUSSIONS

Handling noises. The effect of clusters of noises can be Diminished by two means in Sumblr. First, in tweet stream clustering, noise clusters which are not updated frequently will be deleted as outdated clusters. Second, in the summarization step, tweets from noise clusters are far less likely to be selected into summary, due to their small LexRank scores and cluster sizes. Extension to multi-topic streams. So far we have assumed a tweet stream of only one topic as the input to Sumblr. However, we should note that Sumblr can be easily extended for multi-topic streams. For example, when a new tweet arrives, we first decide its related topics by keyword matching. Then it is delivered into different groups of clusters. Clusters are grouped by their corresponding topical IDs. Consequently, Sumblr is applied within each cluster group. It is important to note that this mechanism allows for distributed system implementation.

5. CONCLUSION

Thus, application will provide a way to generate efficient summary of text and helps in self and social development by providing idea about a topic in a faster way. It also detects human approach depending upon the data collected from various comments and also helps in identifying trends around the world which will helps in business process for decision making.

REFERENCES

[1] P. S. Bradley, U. M. Fayyad, and C. Reina, ***"Scaling clustering algorithms to large databases"***, in Proc. Knowl. Discovery Data Mining, 1998, pp. 915.

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, ***"A framework for clustering evolving data streams"***, in Proc. 29th Int. Conf. Very Large Data Bases, 2003, pp. 8192.

[3] D.Wang, T. Li, S. Zhu, and C. Ding, ***"Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization"***, in Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2008, pp. 307314.

[4] Hull David A. and Grefenstette Gregory. ***"A detailed analysis of English stemming algorithms"***. Rank Xerox ResearchCenter Technical Report.1996.

[5] A. McCallum, and K. Nigam, ***"A comparison of event models for nave Bayes text classi_cation"***, Journal of Machine Learning Research, Vol. 3, 2003, pp. 12651287

[6] T. Zhang, R. Ramakrishnan, and M. Livny, ***"BIRCH: An efficient data clustering method for very large databases"***, in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1996, pp. 103-114.

[7] L. Gong, J. Zeng, and S. Zhang, ***"Text stream clustering algorithm based on adaptive feature selection"***, Expert Syst. Appl., vol. 38,no. 3, pp. 1393-1399, 2011.

[8] J. Zhang, Z. Ghahramani, and Y. Yang, ***"A probabilistic model for online document clustering with application to novelty detection"***, in Proc. Adv. Neural Inf. Process. Syst., 2004, pp. 1617-1624.

[9] G. Erkan and D. R. Radev, ***"LexRank: Graph-based lexical centrality as salience in text summarization"***, J. Artif. Int. Res., vol. 22, no. 1, pp. 457-479, 2004.

[10] Z. He, C. Chen, J. Bu, C. Wang, L. Zhang, D. Cai, and X. He, ***"Document summa- rization based on data reconstruction"***, in Proc. 26th AAAI Conf. Artif. Intell., 2012, pp. 620-626.

[11] N. A. Diakopoulos and D. A. Shamma, ***"Characterizing debate performance via aggregated twitter sentiment"***, in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2010, pp. 1195-1198.

[12] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang, ***"Evolutionary time- line summarization: A balanced optimization framework via iterative substitution"*** in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2011, pp. 745-754