IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 8, August 2016

Automatic Single and Multi Topic Summarization and Evolution to Generate Timeline

Mrs. V. Meenakshi[1], Ms. S. Jeyanthi[2]

Assistant Professor, Department of Computer Science, R.V. Govt. College, Chengalpet, Tamil Nadu, India [1]

M.phil, Research Scholar, Department of Computer Science, Mother Teresa Women's University, Saidapet, Chennai, Tamilnadu, India [2]

Abstract: Nowadays many online social networking services that enable the user to send and receive the information. These information are being shared at an extraordinary rate and their unrefined form, although providing useful information and also be vast. It is difficult for the end users and data analyst to rectify huge amount of noise and data redundancy which included in millions of text. To ease the problem, novel continuous single and multi topic summarization framework has been proposed for text streams. Traditional summarization systems mainly focus on static and small-sized data sets, so, there are not efficient as well as scalable for huge amount of data sets and data streams. Their iterative/recursive results are insensitive to time and difficult to detect topic evolution. Our proposed framework is efficiently designed to deal with dynamic, fast arriving, and large-scale text streams and multi topic summarization. Our framework consists of clustering, single and multi topic summarization and evolution techniques to generate text. A novel clustering algorithm has been proposed to cluster texts and maintain distilled statistics in a data structure. Next a single and multi topic summarization technique has been proposed for generating online summaries and historical summaries of indiscriminate time durations. By comparing manually created summaries and summaries created by some important traditional summarization systems to evaluate the generated summaries efficiently. And finally, an effective topic evolution detection method has been proposed which automatically produce the timelines by monitoring different variations from text streams.

Keywords: Clustering, continuous summarization, single and multi topic summarization, summary, timeline generation, Text stream

I. INTRODUCTION

In recent years, many micro blogging services such as twitter, Weibo, and Tumblr have been developed. Per day 400 millions of priceless tweets[1] for news, blogs, opinions and more have been arrived. Perhaps, for search a particular topic in twitter it might produce millions of tweets, it might be increasing in weeks. Yet if filtering is allowed, searching an important content in tweets might be very terrific, and there might be a chance of occurrence of vast amount of noise and redundancy. New tweets may arrived endlessly at an unpredictable rate that fulfilling the filtering criteria, to make the things very bad.

Summarization is the best possible solution to overcome the information overload problem. Summarization is nothing but that represents a set of documents by a summary which consisting of a number of sentences. Naturally, a good summary should cover the major topics (or subtopics) and have diversity among the sentences to decrease redundancy. Summarization is generally used in the presentation of the content, particularly when users uses the internet with their mobile devices that have little smaller screens compare to the personal desktop computers. Traditional document summarization approaches are not as effective in the large volume of texts as well as the fast and continuous arrival of texts. Text summarization which requires functionalities considerably differ from traditional summarization. In common, text summarization has been taken into consideration the temporal characteristic of the arriving texts.

For example, twitter that contain two interesting features such as an API that permits the user to look for the posts that have the topic phase and a short list of well-liked topic called Trending Topics. An algorithm that has been discussed is used to pick the single post that is representative or is the summary of the number of twitter posts. While the posts returned by the twitter API for a specific topic probably represent several sub-topics or themes, it could be more suitable to create summaries that include the multiple themes rather than just having one post express the whole topic.

Accordingly, a good solution for continuous summarization has to deal with the following three issues:

[1]. https://blog.twitter.com/2013/celebrating-twitter7

Copyright to IJARCCE

DOI 10.17148/IJARCCE.2016.5810

(1) Efficiency—at all times tweet streams are very large in scale, so the summarization algorithm should be highly efficient; (2) Flexibility—Tweet summaries should be provide in a random time durations. (3) Topic evolution—it should automatically detect the changes of sub-topic and the instant that they occur.

II. RELATED WORKS

In this section, review the related work including Data stream clustering, document summarization and timeline detection.

A. Data Stream Clustering

The main idea of the framework for clustering Evolving data streams [1] is to divide the clustering process into two components. The first component is online component which stored the detailed summary statistics in periodically. The second component is offline which uses this summary statistics solitary. For the purpose of efficient and quality of clustering, storage of large volume of data and the fastest usage of this statistical data by using the concept of pyramidal time frame in conjunction with micro-clustering approach.

Q. He et al.[2] proposed a new time-based representation such as bursty features which entirely different from the ancient schemes has been introduced for text streams. It represents the documents dynamically new i.e. representation of document is fully based on its publication date and at any point of time it enlarges the features in relatively to its burstiness. And also it is topic independent. Bursty feature contains two major steps, the first step is to identify the bursty features and the second step is document representation using bursty features/weights.

B. Document Summarization

In this event summarization using tweets [3] disagree with some highly structured and repetitive events such as sports, to summarize the relevant tweet it is necessary to find more sophisticated techniques. Through the Hidden Markov models, to validate the problem of summarizing event-tweets and provide a solution based on learning the fundamental hidden state representation of the event. There are two parts to event summarization, the first one is detecting stages or segments of an event and the second one is summarizing the tweets in each stage.

Chao Shen et al.[4] has been proposed a Participant-based event summarization approach, that identify the participants from the data streams dynamically, then "zooms-in" the twitter event streams to the participant level, distinguish the important sub-events using novel time-content mixture model and generates the event summary increasingly by concatenating the descriptions of the important sub-events. "burstiness" and "cohesiveness" properties has been combined efficiently by mixture model based approach and capture the sub-events effectively otherwise been shadowed by the long tail of other dominant sub-events, producing summaries with considerably better coverage than the state-of-the-art approach. Jie Xu et al.[5] has been introduced the summarization framework for multi-attribute data, that models objects as a set of the equivalent information units and reduce the summarization problem to that of optimizing probabilistic coverage. To overcome the resulting NP-hard problem, highly efficient greedy algorithm has been developed that increase its efficiency through leveraging object-level as well as iteration-level optimization. The proposed framework significantly reaches the high-quality results and also very efficient and scales very well against the size of data set.

C. Multi Document Summarization

A number of notable algorithm has been developed for document summarization that include SumBasic[9] and centroid algorithm[10]. SumBasic's fundamental principle is that words that occur more regularly across documents have a superior chance of being elected for human created multi-document summaries than words that occur less frequently. The centroid algorithm obtains into consideration a centrality measure of a sentence in relation to the overall topic of the document cluster or in relation to a single document summarization.

MEAD [11], is a flexible platform for multi-document multi-lingual publicly existing summarization. MEAD that implements multiple summarization algorithms in addition to provides metrics for evaluating the multi-document summaries.

Many statistical models [12] that is used to analyse the frequency of text and sentences that appear in the first paragraphs. Statistical methods are used in the field of extractive approaches in summarization to merge the heuristics that is used keywords, location and size of sentences, text frequency and topics. Term Frequency-Inverse Document Frequency (TF-IDF) [13] is nothing but a statistical weighting technique that is used to allocate the weight to each term of a document to facilitate returns the term's relevancy in the document. The term frequency component (TF) is used to allocate more and more weight to words that occur regularly within a document because important words are repeated very often. The inverse document frequency component (IDF) makes up for the fact that some words are frequent.

D. Timeline Detection

R. Yan et al. [6] has been proposed Evolutionary Timeline Summarization (ETS), to generate evolution timelines that is similar to our methods. Based on these predefined timestamp sets, the dates of summaries are determined. This system does not generate the timelines dynamically thus; ETS does not meet on the efficiency and scalability issues, which is very important for streaming framework.
Marcus et al. [7] has been developed an algorithm based on the TCP congestion detection. Several systems detect the important moments when the status update volume increased rapidly. Nichols et al. [8] employed a slope-

DOI 10.17148/IJARCCE.2016.5810

based method to find spikes. Later, tweets from each second have been identified and word clouds or summaries are selected. Our method is different from these two approaches to detect topic evolution and produces summaries in an online manner.

III.THE FRAMEWORK FOR SINGLE AND MULTI TOPIC SUMMARIZATION

A novel summarization framework is illustrated in Fig. 1. The framework consists of three main components, the first component is called Clustering module, and the second component is called the High-level summarization module and the third component is Timeline Generation module.

A. Clustering Module

In clustering module, clustering algorithm has been designed, that is used to separate the stream into online component and offline component to maintain the important information of text in clusters. During stream processing are taking into account of conceivable sub-topic delegates and maintained in memory dynamically. The cluster snapshots are organized and stored at different moments by using Pyramidal Time Frame (PTF). Therefore this structure allowing historical tweet data has to be retrieved by any arbitrary time durations.

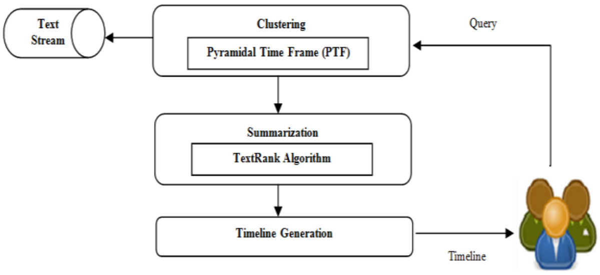


Fig. 2 the Architecture for Timeline Generation

1) Initialization:

At the start of the stream, a small number of tweets have been collected and k-means clustering algorithm is used to create the initial clusters. Next, the corresponding components are initialized; and start the stream clustering process to incrementally update the components whenever a new tweet arrives.

2) Incremental Clustering:

Suppose a new tweet t arrives at time ts, and there are N active clusters at that time. The key problem is to make a decision whether to take up t into one of the current clusters or upgrade t as a new cluster. First find the cluster whose centroid is the closest to new tweet t. specifically, acquire the centroid of each cluster, calculate its cosine similarity to t, and find the cluster C_p with the highest similarity. Note that even though C_p is the closest to tweet t, it does not mean t logically belongs to C_p. The reason is that t may still be very far-away from C_p. In such case, a new cluster should be created.

3) Deleting Outdated Clusters:

For the majority events (such as news, cricket matches and concerts) in tweet streams, timelines is important for the reason that they regularly do not last for an extensive time. Therefore it is secure to delete the clusters representing these sub-topics while they are discussed infrequently. To find out such clusters, a perceptive way is used to estimate the average arrival time (represented as Avg_p) of the final p percent of tweets in a cluster. Though, storing p percent of tweets for each cluster will increase the cost of memory, especially when clusters are raise large.

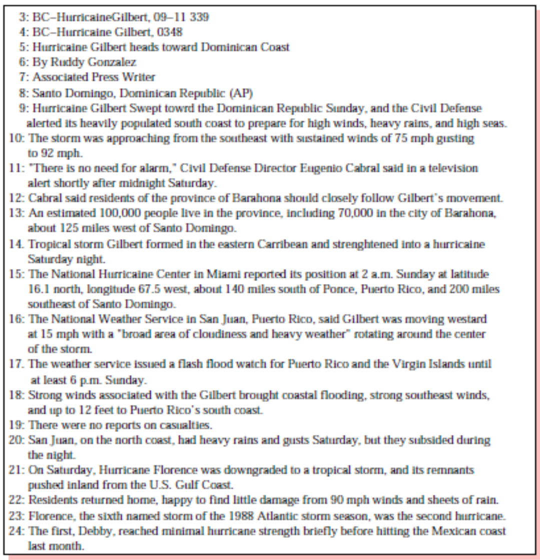
4) Merging Clusters:

If the number of clusters remains increasing with a small number of deletions, system memory will be exhausted. To avoid this, specify an upper limit for the number of clusters as N_{max}. When the upper limit is reached, start the merging process. The process of merging clusters is a greedy manner. First, sort all the cluster pairs by their centroid similarities in a descending order. After that, start to merge two similar pair of clusters. If both clusters are single clusters there is no need to merge with other clusters, they are merged into a new complex cluster. If one of them belongs to the composite cluster (it has been merged with others before), the other cluster is also combined into that composite cluster.

B. Graph-based Extractive Summarization

A graph based extractive summarization algorithm has been proposed for a single document summarization is known as TextRank. TextRank is an unsupervised algorithm. This algorithm is used to extract the most important text or sentences in the document. It does not depending any particular domain or language; it's a domain or language independent. This incredible feature makes this algorithm is broadly used and performs well to automatic summarize the structured text. For automated summarization, TextRank demonstrate any document being graph by means of sentences as nodes. And a function can be used to calculate the resemblance of sentences is required to make the edges between the nodes. This function is used to assign the weights to the graph edges, the higher the similarity between sentences are the more important the edge among them will be in the form of graph.

The text samples and the associated weighted graph constructed for this text as illustrated in Figure 2. The figure also shows sample weights attached to the edges connected to vertex 9^2, and PR formula is used to calculate the final score for each and every vertex and assigned on an undirected graph. The sentences or texts with the highest rank are should be selected for insertion in the abstract. For this sample article, sentences with id-s 9, 15, 16, 18 are has been extracted and resulting in a summary of regarding 100 words, which consistent with automatic evaluation measures, is ranked the second among summaries produced by 15 other systems. Text Rank is used to decide the relation of similarity between two sentences depends on the information that both share. This intersection between the edges is calculated simply as the number of general lexical tokens between them, divided by the length of each to prevent encouraging long sentences. The function included in the novel algorithm can be formalized as:



3: BC–HurricaineGilbert, 09–11 339
4: BC–Hurricaine Gilbert, 0348
5: Hurricaine Gilbert heads toward Dominican Coast
6: By Ruddy Gonzalez
7: Associated Press Writer
8: Santo Domingo, Dominican Republic (AP)
9: Hurricaine Gilbert Swept towrd the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains, and high seas.
10: The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.
11: "There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly after midnight Saturday.
12: Cabral said residents of the province of Barahona should closely follow Gilbert's movement.
13: An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.
14. Tropical storm Gilbert formed in the eastern Carribean and strenghtened into a hurricaine Saturday night.
15: The National Hurricaine Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.
16: The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westard at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.
17. The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.
18: Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds, and up to 12 feet to Puerto Rico's south coast.
19: There were no reports on casualties.
20: San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.
21: On Saturday, Hurricane Florence was downgraded to a tropical storm, and its remnants pushed inland from the U.S. Gulf Coast.
22: Residents returned home, happy to find little damage from 90 mph winds and sheets of rain.
23: Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.
24: The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

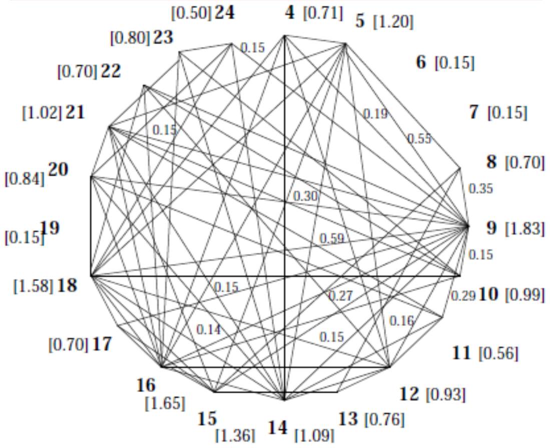


Fig. 2: Sample graph for sentence extraction.

Definition 1. Given Si, Sj are two sentences represented by a set of n words that in Si are represented as $S_i = w_1^i, w_2^i, ..., w_n^i$.

The similarity function for Si, Sj can be defined as:

$$Sim(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)}$$

The result of this process is called a dense graph representing the document. From this graph, PageRank algorithm is used to compute the importance of each and every vertex. The most significant sentences are selected and appeared in the same order as they present in the document as the summary.

BM25: BM25 / Okapi-BM25 are a ranking function generally used for Information Retrieval tasks. BM25 is a difference of the TF-IDF model using a probabilistic model.

Given two sentences R, S, BM25 is defined as:

$$BM25(R,S) = \sum_{i=1}^{n} IDF(s_i) \cdot \frac{f(s_i,R)\cdot(k_1+1)}{f(s_i,R)+k_1\cdot(1-b+b\cdot\frac{|R|}{avgDL})}$$

where k and b are parameters. Taken k = 1.2 and b = 0.75. avgDL is the average length of the sentences in our collection. The text is represented as a weighted graph.

C. Timeline Generation

Text Rank summarization algorithm that is used to find the most ranked sentences or texts from the document. Multi topic summarization is used to extract the information from the multiple sources, and that information is updated to the already existing related topics or otherwise it creates a new topic for that information. Finally real-time timeline is to be generated for single or multi topic of the text which should be top ranked accordingly.

IV. EVALUATION

A. Experimental Setup

The database of the 2002 Document Understanding Conference (DUC) that is used to test the proposed variations as illustrated in Table 1. The corpus has 567 documents that are summarized to 20% of their size. To evaluate results that are used version 1.5.5 of the ROUGE-1, ROUGE-2 and ROUGE-SU4 as metrics as illustrated in figure 3, using a confidence level of 95% and applying stemming. The final result is an average of these three scores. To check the correct performance of our test suite the reference method should be implemented in previously, which extracts the first sentences of each document. The resulting scores of the original algorithm to be equal to those reported in previously: a 2.3% improvement over the baseline.

B. Result

LCS, Cosine Sim, BM25 and BM25+ should be tested in different ways to assign the weight to the edges for the Text Rank graph. The best results were acquired using BM25 and BM25+ with the accurate formula.

TABLE I. EVALUATION RESULTS FOR THE PROPOSED TEXT RANK VARIATIONS

Method	ROUGE-1	ROUGE-2	ROUGE-SU4	Improvement
BM25 ($\varepsilon = 0.25$)	0.4042	0.1831	0.2018	2.92%
BM25+ ($\varepsilon = 0.25$)	0.404	0.1818	0.2008	2.60%
Cosine TF-IDF	0.4108	0.177	0.1984	2.54%
BM25+ (IDF = $\log(N/N_i)$)	0.4022	0.1805	0.1997	2.05%
BM25 (IDF = $\log(N/N_i)$)	0.4012	0.1808	0.1998	1.97%
Longest Common Substring	0.402	0.1783	0.1971	1.40%
BM25+ ($\varepsilon = 0$)	0.3992	0.1803	0.1976	1.36%
BM25 ($\varepsilon = 0$)	0.3991	0.1778	0.1966	0.89%
TextRank	**0.3983**	**0.1762**	**0.1948**	–
BM25	0.3916	0.1725	0.1906	-1.57%
BM25+	0.3903	0.1711	0.1894	-2.07%
DUC Baseline	0.39	0.1689	0.186	-2.84%

The improvement of 2.92% has been achieved in the above original TextRank result using BM25 and ε= 0.25. The following chart that shows the results that are acquired for the different variations that has been proposed. The result of Cosine Similarity was also suitable with a 2.54% improvement above the original method. The 1.40% improvement of LCS variation is also performed better than the original TextRank algorithm. The performance in time was also improved. The 84% of the time needed to process of 567 documents from the DUC2002 database in the original version.

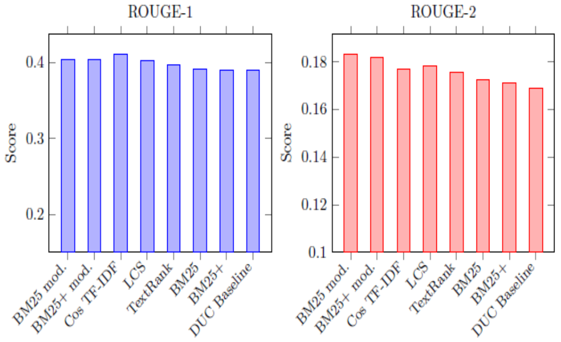


Fig. 3. Scores comparison for ROUGE-1 and ROUGE-2

V. CONCLUSION

A framework has been proposed for single and multi topic which supports the automatic summarization. Clustering algorithm has been proposed to reduce the texts into and maintains them in online manner. A graph based Ranking summarization algorithm has been proposed to generate summaries based on the top ranked approach that contain multiple posts from multiple sources. The grouping of TextRank with BM25 and BM25+, modern Information Retrieval ranking functions that creates a strong system for automatic summarization that execute better than the standard techniques. Finally the topic should be evaluated and generate the timeline automatically. A framework employs these algorithms to produce the timelines for tweet streams. In future, aim to develop a better version of this framework in a distributed system, and calculate it on more complete and large-scale data sets and also image oriented data sets.

REFERENCES

[1] 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. 81–92.

[2] Q. He, K. Chang, E.-P. Lim, and J. Zhang, “Bursty feature representation for clustering text streams,” in Proc. SIAM Int. Conf. Data Mining, 2007, pp. 491–496.

[3] Chakrabarti and K. Punera, “Event summarization using tweets,” in Proc. 5th Int. Conf. Weblogs Social Media, 2011, pp. 66–73.

[4] C. Shen, F. Liu, F. Weng, and T. Li, “A participant-based approach for event summarization using twitter streams,” in Proc. Human Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2013, pp. 1152–1162.

[5] J. Xu, D. V. Kalashnikov, and S. Mehrotra, “Efficient summarization framework for multi-attribute uncertain data,” in Proc. ACM SIGMOD Int. Conf. Manage., 2014, pp. 421–432.

[6] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang, “Evolutionary timeline summarization: A balanced optimization framework via iterative substitution,” in Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2011, pp. 745–754.

[7] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, “Twitinfo: Aggregating and visualizing microblogs for event exploration,” in Proc. SIGCHI Conf. Human Factors Comput. Syst., 2011, pp. 227–236.

[8] J. Nichols, J. Mahmud, and C. Drews, “Summarizing sporting events using twitter,” in Proc. ACM Int. Conf. Intell. User Interfaces, 2012, pp. 189–198.

[9] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, “Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion,” Information Processing & Management, vol. 43, no. 6, pp. 1606–1618, 2007.

[10] D. Radev, S. Blair-Goldensohn, and Z. Zhang, “Experiments in single and multi-document summarization using mead,” DUC-01, vol. 1001, p. 48109, 2001.

[11] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. C¸ elebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang, “Mead – a platform for multidocument multilingual text summarization,” in LREC 2004, Lisbon, Portugal, May 2004.

[12] Das, D., Martins, A.F.T.: A survey on automatic text summarization. Tech. rep., Carnegie Mellon University, Language Technologies Institute (2007)

[13] D. Inouye and J. K. Kalita, “Comparing twitter summarization algorithms for multiple post summaries,” in Proc. IEEE 3rd Int. Conf. Social Comput., 2011, pp. 298–306.