International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 6, Issue 4, April 2017, ISSN: 2278 -7798

A survey on Real-Time Accumulative Short Text Summarization on Comment Streams

[1] N. Vijay Kumar [2] Dr.M.Janga Reddy

1. RESEARCH SCHOLAR JJT UNIVERSITY, CHURELA, JHUNJHUNU, RAJASTHAN .

2. PROF & PRINCIPAL CMRIT KANDLAKOYA, MEDCHAL, HYDERABAD

Abstract: This paper concentrates on the issue of short content rundown on the remark stream of a particular message from informal community administrations (SNS). Since unmistakable clients will ask for the rundown at any minute, existing grouping strategies can't be straightforwardly connected and can't meet the continuous need of this application. In this paper, we show a novel incremental grouping issue for input stream synopsis on SNS. In addition, we propose IncreSTS calculation that can incrementally refresh grouping comes about with most recent approaching remarks continuously. Moreover, we outline an initially representation interface to help clients effectively and quickly get a diagram rundown. From broad exploratory outcomes and a genuine case showing, we confirm that IncreSTS has the benefits of high proficiency, high adaptability, and better taking care of exceptions, which legitimizes the practicability of IncreSTS on the objective issue.

Key words: Real-time short text summarizations, accumulative clustering, comment streams, social network services.

I. INTRODUCTION

Lately, interpersonal organization administrations (SNS) are common and have turned out to be imperative correspondence stages in our everyday life. As indicated by the 2012 insights by the biggest person to person communication site Facebook,1 there are more than 500 million day by day dynamic clients and a normal of 3:2 billion connections (counting Likes and Comments) is produced every day. Arranging the information from 8 big names on Face book, Fig. 1a demonstrates the normal number of remarks in a message. Fig. 1b portrays the pattern of collected number of remarks in a hour for an example message. As can be watched, the amount of remarks is vast, as well as the era rate is surprisingly high. Also, VIPs and partnerships will have high enthusiasm to see how their fans and clients responding to specific themes and substance. With these inspirations, we are propelled to build up a propelled rundown method focusing at remark streams in SNS.

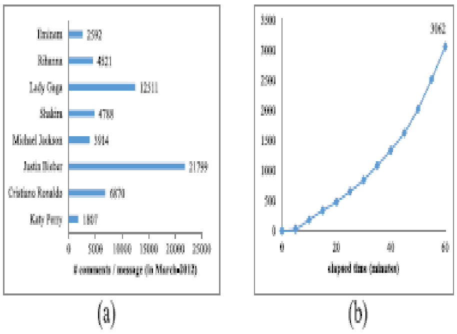


Fig. 1. Simulation results.

In this paper, we don't concentrate on customary remark streams that typically express more entire data, for example, the discourse on items or motion pictures.

Take note of that this issue is obviously not the same as existing examination and has various exceptional attributes and difficulties.

It can be seen that this issue can be displayed as a grouping undertaking. Nonetheless, conventional grouping strategies have a few characteristic confinements that can't be straightforwardly connected here.

In this paper, we investigate the issue of incremental short content outline on remark streams from informal organization administrations. Besides, illustrative remarks in each gathering will likewise be distinguished. Our goal is to create a useful, compact, and great interface that can help clients get a review understanding without perusing all remarks.

Overall, the contributions of this paper can be summarized as follows.

- We model a novel incremental clustering problem based on the requirements of comment stream summarization on SNS.

All Rights Reserved © 2017 IJSETR

- We propose IncreSTS algorithm that can incrementally update clustering results with latest incoming comments in real time.
- We design an at-a-glance presentation, which is concise, informative, and impressive, to help users easily and rapidly get an overview understanding of a comment stream.

2 PRELIMINARIES

In this section, we first give the detailed description of our problem. Then, we present the system model for comment stream summarization on SNS. Related studies are surveyed thereafter in Section 2.2.

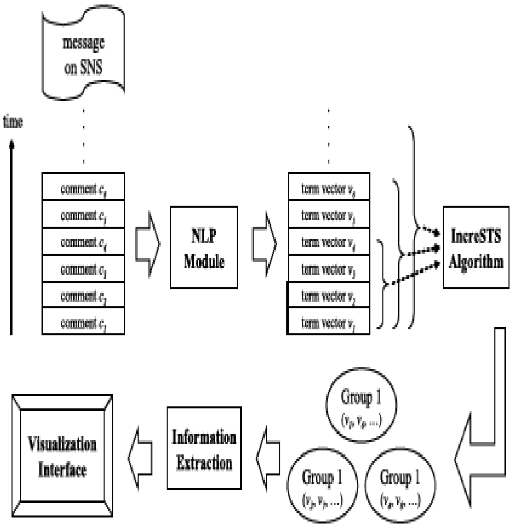


Fig. 2. System model of the proposed framework.

2.1 Problem Description and System Model

We focus on the comment stream added for one message on SNS and aim to generate the immediate summary of comments. The problem we tackle is described as follows.

Problem Description (short text summarization). Given a set of comments S, and the desired number of groups k, find top k groups {C1; C2 ; . . . ; Cj ; . . . , Ck} which have top-k most comments, and the number of comments in Cj is larger than or equal to that of comments in Cjþ1 (i.e., jCjj _ jCjþ1j). Not all comments inS should be included in top-k groups. Moreover, the comments in Cjexpress similar opinions and are a subset of S.

Our main objective is to discover top-k groups where the comments in the same group express similar opinions while the comments belonging to different groups express diverse points of view. The formal definition of the similarity between two comments will be detailed in Section 4.

Once a message is posted on SNS, users can leave comments immediately and the number of comments may rise quickly and continuously. Moreover, readers are usually unwilling to go over the whole list of comments, but they may request to see the summary at any moment. This indicates that the proposed approach should be able to generate the summary result at any time point of a dynamic data stream. To satisfy this requirement, we model this problem as an incremental clustering task. The system model is depicted in Fig. 2.

2.2 Related Works

Owing to the large quantity of user-generated data on SNS, the research topics on alleviating the information overload problem and discovering useful knowledge have attracted much attention recently. In this section, we can broadly classify these works into five categories: 1) Human-assisted mechanisms, 2) Summarization, 3) Rating and filtering, 4) Topic and event detection, and 5) Sentiment analysis. Note social network services are not restricted to well-known social websites, such as Facebook, Twitter, etc.

2.2.1 Human-Assisted Mechanisms

The main notion of this category is to highlight significant comments or provide summary by the assistance of user feedback and judgment.

On the other hand, conducting user survey has also been commonly used, such as in TripAdvisor [5] and HotelClub [4], to obtain detailed grading of respective items. In the literature, for specific services or products, achieving similar summary outcome by the automatic analysis of user comments is investigated in [3] and [7].

2.2.2 Summarization

Regarding the research field of short text summarization, in recent years, numerous works [4], [10], [5], [5], [2] are focused on micro-blogging messages. A variety of techniques have been developed and applied to satisfy different needs of summarization. In [4], a visualization system TwitInfo is presented to enable the convenient browsing of a large collection of Twitter messages (also known as tweets) by detecting and highlighting peaks of highly-discussed activity.

Before the popularity of social network services and micro-blogging websites, blog is one of the primary platforms that users publish content.

On the other hand, the research topic of analyzing product reviews has also attracted much attention [2], [3], [3], [1]. In general, the first step of these approaches is to obtain several aspects of product features from review texts.

2.2.3 Rating and Filtering

Some researchers attempt to relieve the information overload problem by selecting

All Rights Reserved © 2017 IJSETR

representative messages that better express group opinions or contain significant information. The rating mechanism [3], [1], [5], [10], [1] is widely developed to determine the importance of messages.In addition, several types of filtering approaches [5],[4], [2], [1] have also been devised to keep important messages and exclude redundant ones.

Another research direction aims at filtering messages for classification so that users can easily find desired information. Based on author profiles and features within Twitter messages, work of [53] classifies incoming tweets into five categories.

2.2.4 Topic and Event Detection

The key motive of the topic detection on SNS is to help users facilitate the social stream understanding [3], [1], [3], [2]. In [2], the authors propose to generate an entity-based topic profile for each user by examining the entities this user mentions in his/her previous tweets. Works of [3] and [3] aim to develop the topic-based browsing interfaces.

On the other hand, the topic of automatic event detection from large-scale social message streams has attracted much interest as well [11], [5], [5].

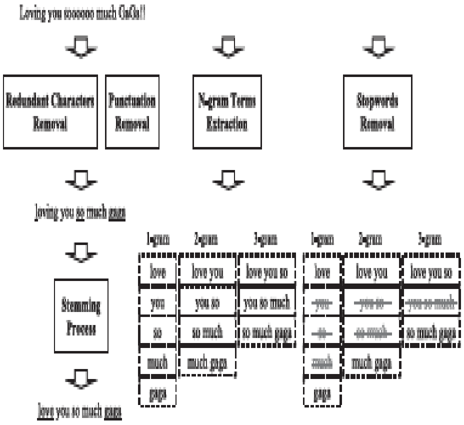


Fig. 3. System model of the proposed framework.

2.2.5 Sentiment Analysis

Many researchers explore the sentiment analysis [4] to discover public mood and emotion hidden in social messages. In general, sentiment classification model is employed to classify messages into pre-defined sentiment labels, such as positive and negative. The system TweetFeel [6] is a realtime Twitter search system that can estimate the ratio of positive to negative tweets mentioning a specific search keyword. In [2], the authors propose to appraise the presidential debate performance via Twitter platfrom according to the numbers of positive and negative tweets.

2.3 Distinguishing Features of IncreSTS

To the best of our knowledge, the proposed IncreSTS is the first fully incremental algorithm aiming to provide immediate and instant summary of real-time social comment streams.

These outcomes indicate that although the proposed method cannot achieve the same overall cluster quality when assessing quality by internal (e.g., minimization of overall distance) and external (human) evaluation, IncreSTS sacrifices cluster quality slightly but can achieve the real-time processing need of the target problem. It is worth mentioning that no comparative clustering methods can meet the requirements of this problem.

In order to meet the real-time requirement of clustering, several researches apply single pass clustering such as [8], [6]. However, in [8], it assumes that all weights of words are determined according to historic news event and will be updated through similar domain knowledge (e.g. CNN, WSJ news). Moreover, in [6], to determine whether a new story is a new event, one of the authors hypotheses is that stories closer together on the stream are more likely to represent related events. It is obvious that this hypothesis cannot be directly applied to our approach since the comments are generated randomly and swiftly.

On the other hand, when compared to existing incremental clustering approaches [4], [2], [4], the major difference is IncreSTS maintains that the radius of each cluster should be smaller than a pre-defined threshold, while other approaches mainly aim to achieve better overall clustering quality, indicating more overhead will be incurred.

3 TERM VECTOR MODEL REPRESENTATION OF COMMENTS

In this section, we elaborate on the details of NLP module that transforms each comment into a set of n-gram terms. Fig. 3 illustrates the procedure flow with an example that demonstrates how to process the comment “Loving you soooooo much GaGa!!”.

The following step is the stemming process. We employ the standard Porter stemming algorithm [4] to reduce inflected and derived words to their stem form (e.g., “loving” is turned into “love” in Fig. 3). Subsequently, the process of n-gram terms extraction is carried out to extract terms that are used for representing this comment. In the example of Fig. 3, n is set to 3, meaning that the comment string will be scanned from left to right to draw out all 1- gram, 2-gram, and 3-gram terms.

It is worth mentioning that we do not intend to concentrate on NLP contributions in this paper. The focus is also not on solving all casual language problems on SNS. Our goal is to employ fundamental processes and develop

All Rights Reserved © 2017 IJSETR

effective heuristics to deal with most cases. Note that more advanced NLP techniques can still be incorporated into our framework when so necessary. In addition, even though the target language in this paper is English, other languages are able to be considered as well while the term vector model representation is applied.

4 DEFINITIONS OF CLUSTERING DETAILS

In this paper, we model the short text summarization as a clustering problem. To meet the practical requirement on SNS and enable the real-time processing, we define a new incremental clustering problem. Detailed definitions are presented in this section.

Consider two comments represented in the term vector model, va ¼ ðt1;a; t2;a; . . . ; tN;aÞ and vb ¼ ðt1;b; t2;b; . . . ; tN;bÞ. Each dimension corresponds to a separate term, and N is the number of dimensions. Moreover, it has been widely observed that text data have directional properties. Thus, we define the modified cosine similarity of two comments as follows:

$$sim(v_a, v_b) = \begin{cases} \frac{v_a \cdot v_b}{D} & \text{if} v_a \cdot v_b \leq D, \\ 1 & \text{if} v_a \cdot v_b > D, \end{cases} \qquad (1)$$

where va _ vb is the inner product of two vectors, and D is a positive integer constant. The denominator of original cosine similarity is the product of the lengths of two vectors. The design of Equation 1 indicates that as long as there exists more than or equal to D common terms in two comments, these two comments can be viewed as with the highest similarity. For instance, if two comments have a mutual 3-gram term, they will also have corresponding mutual sub-terms. Therefore, the value of inner product will be higher.

Definition 1 (Comment-comment distance). The distance between two comments va and vb is defined as:

$$dis(v_a, v_b) = \begin{cases} \frac{1}{sim(v_a, v_b)} - 1 & \text{if } sim(v_a, v_b) \neq 0, \\ \infty & \text{if } sim(v_a, v_b) = 0. \end{cases} \qquad (2)$$

With such definition, when va and vb have more than or equal to D common terms (i.e., simðva; vbÞ ¼ 1), the distance between them is zero.

Definition 2 (Center of cluster). Let fv1; v2; . . . ; vi; . . . ; vng be the set of comments belonging to cluster Ce , where vi ¼ ðt1;i; t2;i; . . . ; tj;i; . . . ; tN;iÞ. The center vce of cluster Ce is defined as:

$$vc_e = (tc_{1,e}, tc_{2,e}, \ldots, tc_{j,e}, \ldots, tc_{N,e}), \qquad (3)$$

$$tc_{j,e} = \sum_{p=1}^{n} t_{j,p} \quad . \qquad (4)$$

Similarly, the vector of cluster center is not normalized. In addition, each element in vce is not averaged by the number of total comments in the cluster since such an operation is not necessary for distance calculation.

Definition 3 (Comment-cluster distance). The similarity between comment vi and cluster Ce (whose center is vce) is defined as:

$$sim(v_i, C_e) = \begin{cases} \frac{f(v_i, C_e)}{T} & \text{if } f(v_i, C_e) \leq T, \\ 1 & \text{if } f(v_i, C_e) > T, \end{cases} \qquad (5)$$

$$f(v_i, C_e) = \sum_{p=1}^{N} \begin{cases} 2 & \text{if } t_{p,i} \times tc_{p,e} > 2, \\ t_{p,i} \times tc_{p,e} & \text{otherwise.} \end{cases} \qquad (6)$$

Accordingly, the distance between comment vi and cluster Ce is defined as:

$$dis(v_i, C_e) = \begin{cases} \frac{1}{sim(v_i, C_e)} - 1 & \text{if } sim(v_i, C_e) \neq 0, \\ \infty & \text{if } sim(v_i, C_e) = 0. \end{cases} \qquad (7)$$

In Equation (5), T is a positive integer constant. Note that the effect of function fðvi; CeÞ is analogous to the inner product of the comment vi and the center vce of cluster Ce.

The difference is the product value in each dimension is limited to 2. This design is to avoid the bias caused by certain terms with large counts in the cluster. It is favorable to group comments with more mutual terms.

Definition 4 (Cluster-cluster distance). The distance between two clusters is defined as the distance between two cluster centers derived from Equation (2).

Definition 5 (Radius of cluster). The radius ra of cluster Ca is defined as the farthest distance between the center of Ca and any comment in this cluster.

Based on the perspective of clustering problem, the short text summarization task is defined as follows.

Definition 6 (Short text summarization on comment streams). Given a set of comments S, and a desired number of cluster k, find top-k clusters {C1; C2; . . . ; Cj ; . . . ; Ck} which have top-k most comments, and the number of comments in Cj is larger than or equal to that in Cjþ1 (i.e., jCjj _ jCjþ1j). Not all comments in S should be included in top-k clusters, and Cj _ S. In addition, the radius of each cluster should be smaller than the radius threshold ur.

By limiting the radius of each cluster, we can ensure that the comments in the same cluster express similar opinions. Furthermore, to enable the real-time processing, which cannot be realized by most existing clustering methods, we propose an incremental algorithm that is specifically designed for short text summarization on comment streams and is introduced in the next section.

5 INCREMENTAL SHORT TEXT SUMMARIZATION

In this section, we aim to develop efficient approaches in discovering top-k groups of opinions towards a specific message on SNS. A batch version of short text summarization algorithm is first introduced in Section 5.1. We then propose a fully incremental algorithm in Section 5.2. Finally, the design of visualization interface, including key-term cloud presentation and representative comments extraction,
is presented in Section 5.3.

5.1 BatchSTS Algorithm: Batch Version

According to the problem definition of Definition 6, we propose the algorithm BatchSTS that is the batch version for solving this problem.

There are two main steps in BatchSTS. The aim of the first step, shown in lines 2-11 of Algorithm 1, is to find all connected components of the comment set S. The points belonging to the same connected component will be merged as a cluster. It can be imagined that there will be a link between two comments as their distance is not infinite. Meanwhile, in line 17, itwill be checkedwhether vj can be merged with other excluded comments. After this step, all clusterswill satisfy the radius restriction, and finally, BatchSTS outputs the top-k clusters with top-k most comments.

```
Algorithm BatchSTS
Input: S: the comment set
r: the radius threshold
Output: top-k clusters which have top-k most comments
1. Initialize C = Ø;
2. for each element vi of S
3.   if there exists any cluster Cj where dis(vi,Cj) is not infinite
4.     Add vi into anyone of these clusters;
5.   else
6.     Form a new cluster Cnew with the comment vi;
7.     C = C Cnew;
8. for each non-single-point element Ci of C
9.   for each non-single-point element Cj of C where i j
10.    if dis(Ci,Cj) is not infinite
11.      Merge Ci and Cj;
12. for each non-single-point element Ci of C
13.   while the radius of Ci is larger than or equal to r
14.     for each comment vj in Ci
15.       if dis(vj,Ci) r
16.         Exclude vj from Ci;
17.         Check whether vj can be merged with other excluded comments;
18. Output top-k clusters in C which have top-k most comments;
End
```

5.2 IncreSTS Algorithm: Incremental Version

Due to the high popularity of SNS, the number of comments for a specific message may increase very quickly, and users will request to view the summary of comments at any time.

Moreover, since new messages appear continuously, users generally only view the summary of a specific message once and will not go back to browse the updated summary in the future.

```
Algorithm IncreSTS
Input: C: the set of previous clustering result
vnew: the newly-incoming comment
r: the radius threshold
Output: top-k clusters which have top-k most comments
1. Ca = {Ci | Ci is an element of C dis(vnew,Ci) is not infinite};
2. Cb= {Cj | Cj is an element of Ca dis(vnew,Cj) < r};
3. if Cb is not empty
4.   Add vnew into Cadded which have most comments in Cb;
5.   Initialize Cchanged = Ø;
6.   for each element Ci of Ca where Ci Cadded
7.     for each comment vj in Ci
8.       if dis(vj,Cadded) < r
9.         Add vj into Cadded;
10.        Exclude vj from Ci;
11.        Cchanged = Cchanged Ci;
12.  for each element Ci of Cchanged
13.    while V = {vj | dis(vj,Ci) r} is not empty
14.      Exclude all elements in V from Ci;
15.      Try to add each comment in V into other clusters
         from large to small sizes;
16. else
17.   Form a new cluster Cnew with the comment vnew;
18.   C = C Cnew;
19. Output top-k clusters in C which have top-k most comments;
End
```

Three main steps are involved in IncreSTS. Initially, we have to find the cluster which the newly-incoming comment vnew should be added into. In line 1 of Algorithm 2, the distances between vnew and all existing clusters are calculated.

Lemma 1. When a comment is added into a cluster C0, no comments in C0 will thus be excluded.

Lemma 2. When a comment is excluded from a cluster C0, some comments in C0 may thus be excluded.

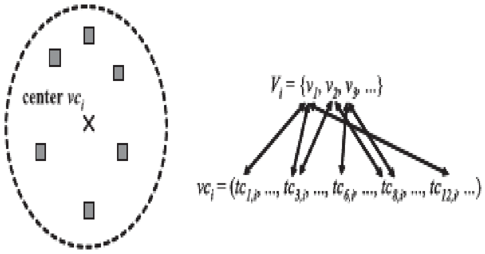

center vc_i
X
$V_i = \{v_1, v_2, v_3, ...\}$
$vc_i = (tc_{1,i}, ..., tc_{3,i}, ..., tc_{6,i}, ..., tc_{8,i}, ..., tc_{12,i}, ...)$

Fig. 4. An illustration of the designed data structure for a cluster.

For Lemma 1, since adding a comment into a cluster will only cause the values in some dimensions of the cluster center to be increased (but not decreased), the similarity (defined in Equation (5)) between each original comment and this cluster is only possible to be increased. This indicates the distance is also guaranteed to be smaller than the radius threshold. In contrast, for Lemma 2, excluding a comment from a cluster will cause the values in some dimensions of the cluster center to be decreased, and therefore the similarities between some original comments and this cluster are possible to be decreased.

Regarding the efficiency issue of IncreSTS, the complexity of the first step is OðmÞ, where m is the number of existing clusters. For instance, as shown in Fig. 4, comment v2 has term 3 and term 8. Most importantly, these pointers enable the capability of deriving the inverted index of terms. For instance, in Fig. 4, term 3 appears in comment v1 and comment v2.

5.3 Visualization Interface

After the top-k clusters are generated, the next issue is how to present the summarization results. On SNS with the exploding amount of information, we aim to provide a concise and at-a-glance visualization interface that enables users to quickly get an overview understanding of a comment stream.

```
Procedure Key-Term Extraction
Input: vc: the center of a cluster
%: the threshold of overlapping percentage
Output: Skey-terms: the set of representative key-terms
1. Initialize Skey-terms = vc;
2. for each set of n-gram terms in Skey-terms
3. Eliminate the terms whose counts do not rank top k in this set;
4. for each term ti in Skey-terms
5. if there exists any term tj where (tj.ngram == ti.ngram && tj.count >= ti.count)
6. if there are over % of words in ti also contained in tj
7. Eliminate ti from Skey-terms;
8. for each term ti in Skey-terms
9. if there exists any term tj where (tj.ngram > ti.ngram)
10. if there are over % of words in ti also contained in tj
11. Eliminate ti from Skey-terms;
12. Output the set Skey-terms of representative key-terms;
End
```

Regarding the first part, Algorithm 3 shows the algorithmic form of the proposed Key-Term Extraction procedure. An intuitive way is to extract the top-k terms with the k most frequency counts from the cluster center. For instance, it is not necessary to retain both the terms “love you so much” and “I love you”. The details of this elimination process consist of two steps. First, we examine each set of n-gram terms respectively.

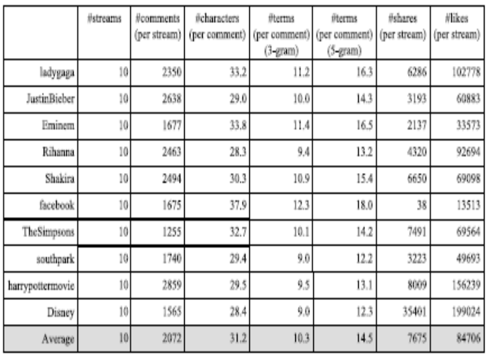
Example 1. Given the threshold u percent of overlapping percentage equal to 50 percent and two terms “I love you” (count ¼ 20) and “love you gaga” (count ¼ 10), the term “love you gaga” will be eliminated since its count is smaller than that of the term ”I love you”, and there are 66:7 percent (2=3) of words contained in the first term.

Finally, the terms in the output set Skey_terms will be used to form the key-term cloud. Note that the display size of each key-term is proportional to the frequency counts in the cluster center.

6 PERFORMANCE EVALUATION

We conduct extensive experiments with comment data streams collected from Facebook to evaluate the performance of IncreSTS.

TABLE 1
Detailed Information of Comment Streams Collected from Facebook



	#streams	#comments (per stream)	#characters (per comment)	#terms (per comment) (3-gram)	#terms (per comment) (5-gram)	#shares (per stream)	#likes (per stream)
ladygaga	10	2350	33.2	11.2	16.3	6286	102778
JustinBieber	10	2638	29.0	10.0	14.3	3193	60883
Eminem	10	1677	33.8	11.4	16.5	2137	33573
Rihanna	10	2463	28.3	9.4	13.2	4320	92694
Shakira	10	2494	30.3	10.9	15.4	6650	69098
facebook	10	1675	37.9	12.3	18.0	38	13513
TheSimpsons	10	1255	32.7	10.1	14.2	7491	69564
southpark	10	1740	29.4	9.0	12.2	3223	49693
harrypottermovie	10	2859	29.5	9.5	13.1	8009	156239
Disney	10	1565	28.4	9.0	12.3	35401	199024
Average	10	2072	31.2	10.3	14.5	7675	84706

6.1 Experimental Design

All Rights Reserved © 2017 IJSETR

We collect real comment streams from Facebook through Facebook Graph API.3Among the top 25 Facebook pages (with most number of likes) in September 2012,4 we choose 10 of them, and for each page, 10 social messages having from 1;000 to 3;500 comments are retrieved. Table 1 summarizes the detailed information of this dataset comprising 100 comment streams totally.

To validate the performance of the proposed methods, several representative clustering algorithms are considered for comparison, including DBSCAN (density-based), KMeans (partition-based), and AGNES (hierarchical).

6.2 Efficiency Issues

In this section, the efficiency and effectiveness issues are investigated. First, we attempt to answer the question: "Can IncreSTS meet the real-time need of short text summarization on SNS?" Four representative clustering methods are included to compare with BatchSTS and IncreSTS. Fig. 5a shows the performance of execution time with the number of comments varied. shown in Fig. 5a is the total time of processing all data points.

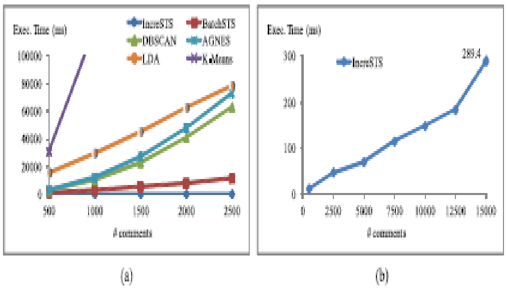


Fig. 5. (a) Comparison of efficiency performance. (b) Scalability of IncreSTS.

We can observe that K-Means (K is set to 5) has the worst performance, consuming over 30 seconds in clustering 500 comments.

Fig. 5b further depicts the results of the scalability experiment of IncreSTS. We can see that the required update time increases as the number of existing comments grows.

7 CONCLUSION

In this paper, to enable the capability of comment stream summarization on SNS, we model a novel incremental clustering problem and propose the algorithm IncreSTS, which can incrementally update clustering results with latest incoming comments in real time. With the output of IncreSTS, we design a visualization interface consisting of basic information, key-term clouds, and representative comments. This at-a-glance presentation enables users to easily and rapidly get an overview understanding of a comment stream. From extensive experimental results and a real case demonstration, we verify that IncreSTS possesses the advantages of high efficiency, high scalability, and better handling outliers, which justifies the practicability of IncreSTS on the target problem.

REFERENCES

[1] Amazon [Online]. Available: http://www.amazon.com/, 2014.

[2] Experimental Demo Page [Online]. Available: http://140.109.21.214/public/IncreSTS/index.htm, 2014.

[3] Facebook [Online]. Available: http://www.facebook.com/, 2014.

[4] HotelClub [Online]. Available: http://www.hotelclub.com/,2014.

[5] TripAdvisor [Online]. Available: http://www.tripadvisor.com/,2014.

[6] TweetFeel [Online]. Available: http://www.tweetfeel.com/,2014.

[7] YouTube [Online]. Available: http://www.youtube.com/, 2014.

[8] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang,"On-line new event detection and tracking," in Proc. 21th Annu.Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 1998, pp. 37–45.

[9] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics:Ordering points to identify the clustering structure," in Proc.ACM SIGMOD Int. Conf. Manag. Data, 1999, vol. 28, no. 2,pp. 49–60.

[10] S. Baccianella, A. Esuli, and F. Sebastiani, "Multi-facet Rating of Product Reviews," in Proc. 31st Eur. Conf. IR Res. Adv. Inf. Retrieval,2009, pp. 461–472.

[11] H. Becker, M. Naaman, and L. Gravano, "Learning similarity metrics for event identification in social media," in Proc. 3rd ACM Int.Conf. Web Search Data Mining, 2010, pp. 291–300.

[12] H. Becker, M. Naaman, and L. Gravano, "Selecting quality twitter content for events," in Proc. 5th Int. AAAI Conf. Weblogs Social Media, 2011, pp. 442–445.

[13] M. S. Bernstein, B. Suh, L. Hong, J. Chen, S. Kairam, and E. H. Chi,"Eddi: Interactive topic-based browsing of social status streams,"in Proc. 23nd Annu. ACM Symp. User Interface Softw. Technol., 2010,pp. 303–312.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," J. Mach. Learning Res., vol. 3, pp. 993–1022, 2003.

[15] J. Bollen, H. Mao, and A. Pepe, "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena," in Proc.5th Int. AAAI Conf. Weblogs Social Media, 2011, pp. 450–453.

All Rights Reserved © 2017 IJSETR

Mr. N.Vijay Kumar, received B.Tech(CSE) from JNTUH, M.Tech(NN) from JNTUK. And Pursuing Phd from JJTU Rajastan. My Research interest is Data ware housing and Data mining.

Dr.M Janga Reddy Principal CMRIT Hyderabad, His Research Interest is Data ware housing and Data mining, Network Security and software engineering.

All Rights Reserved © 2017 IJSETR