IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

Increst: Incremental Short Text Summarization On Comment Streams From Social Network Services

Jaseela Mahamood
MTech Scholar, LBS College of engineering Kasaragod, Kerala, India

Abstract: In this paper we are focusing on the problem of short text summarization on the comment stream of a specific message from social network services (SNS). The quantity of comments may increase at a high rate after a social message is published, due to the high popularity of SNS.Then users may desire to get a brief understanding of a comment stream without reading the whole comment list, we attempt to group comments with similar content together and generate a concise opinion summary for this message. Since distinct users will request the summary at any moment, existing clustering methods cannot be directly applied and cannot meet the real-time need of this application. In this paper, we model a novel incremental clustering problem for comment stream summarization on SNS. Moreover, we propose IncreSTS algorithm that can incrementally update clustering results with latest incoming comments in real time. Furthermore, we design an at-a-glance visualization interface to help users easily and rapidly get an overview summary .IncreSTS possesses the advantages of high efficiency, high scalability, and better handling outliers.

Keywords—real-time short text summarization, incremental clustering, comment streams, social network services

I. INTRODUCTION

Social network services (SNS) have become important communication platforms in our daily life. Note that for each message, users are able to express their opinions by forwarding, giving a like, and leaving comments on it. As can be observed, not only the quantity of comments is large, but also the generation rate is remarkably high.Users unnecessarily and almost impossibly go over the whole comment list of each message. However, we may still desire to know what are they talking about and what are the opinions of these discussion participants.Moreover, celebrities and corporations will have high interest to understand how their fans and customers reacting to certain topics and content. With these motivations, we are inspired to develop an advanced summarization technique targeting at comment streams in SNS. Numerous studies and systems have proposed techniques and mechanisms to generate various types of summaries on comment streams. One major category aims to extract representative and significant comments from messy discussion. We target at comment streams in SNS that are in short text style with casual language usage. For each social message, our main objective is to cluster comments with similar content together and generate a concise opinion summary for this message. We want to discover how many different group opinions exist and provide an overview of each group to make users easily and rapidly understand. when a person uploads a photo to SNS, there are hundreds and thousands of comments given by her fans during a short period of time. Therefore, our goal is developing an efficient and effective technique to identify the clusters of these comments. Note that this problem is clearly different from existing research and possesses numerous unique characteristics and challenges. First, the quantity of comments may increase at a high rate right after a social message is published. Moreover, distinct users will request the summary result at any moment. For these reasons, inorder to immediately generate a summary based on the current comment stream, an incremental approach is preferable to meet the real-time needs of this application .Furthermore, the comments in SNS are usually short ,and users widely make use of informal and unstructured texts that contain acronyms, shortening words, etc. This phenomenon increases the difficulty of determining the similarity between comments. On the other hand, it is worth mentioning that instead of emphasizing on the quality of clustering, the most crucial point of this task is to produce a general summary promptly so that users can easily get the overview of a comment stream .It can be perceived that this problem is able to be modeled as a clustering task. However, traditional clustering methods have several inherent restrictions that cannot be directly applied here. First, the computational complexity of existing methods is high, and they cannot straightforwardly adapt to satisfy the incremental need. Moreover, in this problem, defining the number of desired clusters in advance is unreasonable, which is required in many clustering algorithms. In addition, there will be a lot of outliers in a comment stream, meaning that without employing a good strategy for selecting initial cluster centers, existing method may be prone to poor resultsIn this paper, we explore the problem of incremental short text summarization on comment streams from social network services. We model this problem as an incremental clustering task and propose the IncreSTS (standing for Incremental Short Text Summarization) algorithm to discover the top-k clusters including different groups of opinions towards one social message. For each comment cluster, important and common terms will be extracted to construct a key-term cloud. This key-term cloud provides an at-a-glance presentation that users can easily and rapidly understand the main

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

points of similar comments in a cluster. Moreover, representative comments in each group will also be identified. Our objective is to generate an informative, concise, and impressive interface that can help users get an overview understanding without reading all comments .With employing some basic NLP procedures, each comment is transformed to a set of n-gram terms. On the other hand, we define new similarity measures to adequately determine the distance between comments and clusters. According to the new definitions, we propose a fully incremental algorithm that is almost parameter free and can handle the outlier problem. Furthermore,the most significant advantage of our algorithm is its high efficiency, indicating that it can generate clustering results with latest incoming comments in real time. These capabilities certainly meet the need of comment stream summarization on SNS

II. RELATED WORK

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 addition to the comment stream data discussed in this paper, previous works also target at different types of social data and explore various research topics related to solving the information overload problem on SNS. . Regarding the research field of short text summarization, in recent years, numerous works [6][8] are focused on micro-blogging messages. A variety of techniques have been developed and applied to satisfy different needs of summarization. In [8], 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. Another map-like presentation system TwitterStand further considers incorporating geographic location of tweets to automatically obtain late breaking news. In addition, with collections of short posts on a specific topic, the authors in [9] aim to create short summary sentences that best describe the primary gist of what users are saying about. With similar intention of [3], work of [4] employs both generative model and user behaviour model to synthesize content from micro-blogging messages on the same topic into a prose description of fixed length. Before the popularity of social network services and micro-blogging websites, blog is one of the primary platforms that users publish content. As for the summarization of traditional blogs, one main research direction is to extract and discover representative sentences The authors in [10][11]consider utilizing user feedback comments to identify important sentences on a blog post. Social contexts are included to model the framework, and the mutual reinforcement between Web documents and their associated social data is employed to generate summaries. On the other hand, the research topic of analysing product reviews has also attracted much attention [10] In general, the first step of these approaches is to obtain several aspects of product features from review texts. Subsequently, in addition to traditional techniques of data mining or machine learning, natural language processing and sentiment analysis are commonly incorporated to achieve various summarization needs. Furthermore, the authors in [9] propose to extract descriptions of problems related to specific products or services from micro-blogging data. Note that different from existing methods for the automatic analysis of product reviews, the content of social comment streams, which is the target in this paper, is often much shorter and briefer.

III. PROBLEM DESCRIPTION AND SYSTEM MODEL

In this section, we first give the detailed description of our problem. Then, we present the system model for comment stream summarization on SNS. 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 Cj+1 Not all comments in S should be included in top-k groups. Moreover, the comments in Cj express similar opinions and are a subset of S.

System model of the proposed framework.

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. 1.IncreSTS (Incremental Short Text Summarization) algorithm to discover the top-k clusters including different groups of opinions towards one social message. For each comment cluster, important and common terms will be extracted to construct a key-term cloud. Thiskey-termcloudprovides an at-a-glance presentation that users can easily and rapidly understand the main points of similar comments in a cluster.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

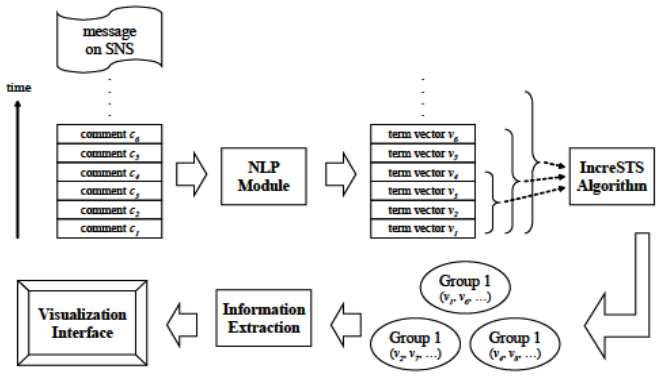

Fig. 1: System model of the proposed framework.

We adopt the term vector model, and therefore each comment is transformed into a set of n-gram terms by the NLP module. Since informal and unstructured texts are widely used on SNS, we also apply some heuristics to enhance the quality of n-gram terms that can better represent each comment. In such a context, whenever a request is received, the proposed IncreSTS algorithm will efficiently produce top-k groups of opinions in real time. It can be perceived that it is infeasible to repeatedly carry out the complete clustering task due to the high complexity. For this reason, we design the IncreSTS algorithm in incremental manner, meaning that the clustering result of the previous phase will be leveraged to generate the current summary with newly-incoming comments. Finally, for the visualization interface, representative terms will be extracted to form a key-term cloud for each group. Thus, users will be provided a concise, informative, and at-a-glance presentation that can help them easily comprehend the main points of responses to one message on SNS

IV.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 ngram terms

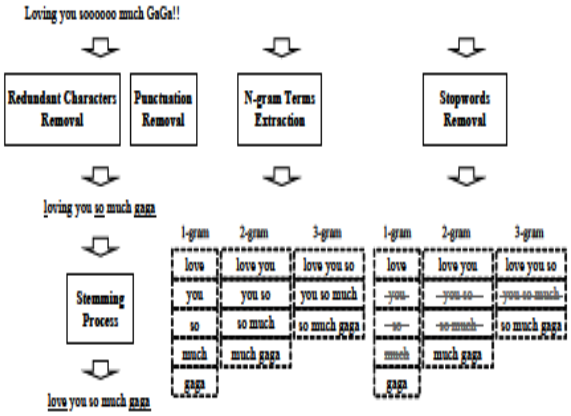

Fig. 2: System model of the proposed framework.

. Fig. 2 illustrates the procedure flow with an example that demonstrates how to process the comment ”Lovingyou soooooo much GaGa!!”.Initially, for each word, the process of punctuation removal will be applied to eliminate unnecessary punctuation marks connected with this word. In the example of Fig. 2, 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. Finally, stop words removal process is executed to delete terms entirely composed of stop words. Note that as long as there is at least one non-stop word appearing in a term, this term will be viewed as a valid one

IV.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 the Section (1) We then propose a fully incremental algorithm in Section (2) Finally, the design of visualization interface, including key-term cloud presentation and representative comments extraction, is presented in Section (3)

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

(1) BatchSTS Algorithm: Batch Version

According to the problem definition of Definition we propose the algorithm BatchSTS that is the batch version for solving this problem. The algorithmic form of BatchSTS is outlined in Algorithm 1. BatchSTS takes the whole comment set S as the input. The second input is the radius threshold θr used for determining how similar the comments are in a cluster. There are two main steps in BatchSTS. The aim of the first step, 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. for each comment vi of S, we examine whether there is any existing cluster Cj where dis(vi;Cj) is not infinite. If there is, this comment will be added into anyone of these clusters. Otherwise, a new single-point cluster is formed with the comment vi., we calculate the distances between two centers of any pair of non-single-point clusters, and if the distance between two clusters is not infinite, they will be merged together. the objective of the second main step is to guarantee the radius of each cluster is smaller than the threshold θr. To meet this requirement, for each non-single-point cluster Ci, we exclude the comment vj where dis(vj ;Ci) is larger than or equal to θr. it will be checked whether vj can be merged with other excluded comments. After this step, all clusters will satisfy the radius restriction, and finally, BatchSTS outputs the topk clusters with top-k most comments.

Algorithm BatchSTS
Input: S: the comment set
θ_r: the radius threshold
Output: top-k clusters which have top-k most comments
1. Initialize $C = \emptyset$;
2. **for** each element v_i of S
3. **if** there exists any cluster C_j where $dis(v_i, C_j)$ is not infinite
4. Add v_i into anyone of these clusters;
5. **else**
6. Form a new cluster C_{new} with the comment v_i;
7. $C = C \cup C_{new}$;
8. **for** each non-single-point element C_i of C
9. **for** each non-single-point element C_j of C where $i \neq j$
10. **if** $dis(C_i, C_j)$ is not infinite
11. Merge C_i and C_j;
12. **for** each non-single-point element C_i of C
13. **while** the radius of C_i is larger than or equal to θ_r
14. **for** each comment v_j in C_i
15. **if** $dis(v_j, C_i) \geqq \theta_r$
16. Exclude v_j from C_i;
17. Check whether v_j can be merged with other excluded comments;
18. Output top-k clusters in C which have top-k most comments;
End

Algorithm 1. Algorithmic form of BatchSTS.

(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. In such a context, to immediately produce the latest top-k clusters, we propose the IncreSTS algorithm that has the capability of incremental update. The primary concept of IncreSTS is to maintain the clustering result of the previous phase, and to incrementally update the clustering result with the newly-incoming comment. Algorithm 2 outline algorithmic form of inreSTS. Three main steps are involved in IncreSTS. Initially, we have to find the cluster which the newly-incoming comment vnew should be added into.The distances between vnew and all existing clusters are calculated. Among the clusters (in the set Cb) whose distances are smaller than θr, we choose the cluster Cadded that has most comments. It is worth noting that vnew is not added into the cluster where dis(vnew;Cj) is the smallest. The second main step is to examine whether comments in other clusters can be absorbed into Cadded. Instead of checking all clusters, only the cluster Ci in Ca, where dis(vnew;Ci) is not infinite, should be examined. If dis(vnew;Ci) is infinite, it means there exists no common term between vnew and the center of Ci. Thus, the comments in Ci are impossible to be absorbed into Cadded. when some comments are added into Cadded Nevertheless, for the clusters where some comments have been absorbed into Cadded, we need to check whether the radius restriction is still satisfied. This checking process, shown in lines 12-15.If some comments are excluded, they will be attempted to be

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

added into other clusters from large to small sizes under the radius restriction. Finally, IncreSTS outputs the top-k clusters with top-k most comments.

Algorithm IncreSTS

Input: C: the set of previous clustering result
v_{new}: the newly-incoming comment
θ_r: the radius threshold

Output: top-k clusters which have top-k most comments

1. $C^a = \{C_i \mid C_i$ is an element of $C \cap dis(v_{new}, C_i)$ is not infinite$\}$;
2. $C^b = \{C_j \mid C_j$ is an element of $C^a \cap dis(v_{new}, C_j) < \theta_r\}$;
3. **if** C^b is not empty
4. Add v_{new} into C_{added} which have most comments in C^b;
5. Initialize $C_{changed} = \emptyset$;
6. **for** each element C_i of C^a where $C_i \neq C_{added}$
7. **for** each comment v_j in C_i
8. **if** $dis(v_j, C_{added}) < \theta_r$
9. Add v_j into C_{added};
10. Exclude v_j from C_i;
11. $C_{changed} = C_{changed} \cup C_i$;
12. **for** each element C_i of $C_{changed}$
13. **while** $V = \{v_j \mid dis(v_j, C_i) \geqq \theta_r\}$ is not empty
14. Exclude all elements in V from C_i;
15. Try to add each comment in V into other clusters from large to small sizes;
16. **else**
17. Form a new cluster C_{new} with the comment v_{new};
18. $C = C \cup C_{new}$;
19. Output top-k clusters in C which have top-k most comments;

End

Algorithm 2. Algorithmic form of IncreSTS.

(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. Therefore, two types of summaries are considered. For each cluster, a set of representative key-terms that are frequently mentioned will be extracted to construct a key-term cloud. Moreover, several representative comments will also be identified. Algorith 3 shows the algorithmic form of the proposed key term extraction.Anntuitive way is to extract the top-k terms with the k most frequency counts from the cluster center. However, such strategy will lead to the problem that 1- gram terms dominate over n-gram terms, where n is an integer and larger than or equal to 2. To remedy this defect, in each set of n-gram terms, top-k terms with k most counts are extracted, 3. Subsequently, the second step is to eliminate the terms which have overlapping words with others, indicating that the terms with repeated meaning will be excluded. For instance,The details of this elimination process consist of two steps. First, we examine each set of n-gram terms respectively. For each term ti in a set, ti will be eliminated if there exists another term tj whose count is larger than or equal to that of this term, and there are over θ% of words in ti also contained in tj , where θ% is the threshold of overlapping percentage.

IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
NCRTET-17
National Conference on Recent Trends in Engineering & Technology
Government Polytechnic College Kasaragod, Kerala
Vol. 4, Special Issue 1

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

Algorithm 3. Algorithmic form of procedure Key-Term Extraction.

V.CONCLUSION AND FUTURE WORK

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. IncreSTS possesses the advantages of high efficiency, high scalability, and better handling outliers.As a future work we plan to apply sentiment analysis on comment stream. Sentiment Analysis is the process of determining whether a piece of writing is positive or negative. It's also known as opinion mining, deriving the opinion or attitude of a speaker.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. The Journal of Machine Learning Research, 3:993–1022, 2003.
[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. Proc. of the 2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '96)
[3] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108,
[4] Clustering Data Streams Based on Shared Density between Micro-Clusters Michael Hahsler, Member, IEEE and Matthew Bola~nos
[5] Density Micro-Clustering Algorithms on Data Streams: A ReviewAmineh Amini, Teh Ying Wah
[6] J.-Y. Weng, C.-L. Yang, B.-N. Chen, Y.-K. Wang, and S.-D. Lin. IMASS: An Intelligent Microblog Analysis and Summarization System.
[7] A Survey On Short Text Summarization Of Comment Streams On Social Network Sites Ms. Pooja S.Choudhari , Prof. S. S. Nandgaonkar
[8] Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. TwitInfo: Aggregating and Visualizing Microblogs for Event Exploration. Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems 2011.
[9] B. Sharifi, M.-A. Hutton, and J. K. Kalita. Experiments in Microblog Summarization. Proc. of the 2nd IEEE International Conference on Social Computing
[10] M. Hu, A. Sun, and E.-P. Lim. Comments-Oriented Blog Summarization by Sentence Extraction. Proc. Of the 16th ACM InternationalConference on Information and Knowledg Management (CIKM'07), 2007.
[11] M. Hu, A. Sun, and E.-P. Lim. Comments-Oriented Document Summarization: Understanding Documents with Readers' Feedback. Proc. of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval, 2008.