Realtime Event Summarization from Tweets with Inconsistency Detection

Lingting Lin, Yunjie Wang, Chen Lin

Department of Computer Science, Xiamen University chenlin@xmu.edu.cn

Abstract. The overwhelming amount of event relevant tweets highlights the importance of realtime event summarization systems. Previous studies couldn't guarantee the integrity of a realtime summary. When new information emerges, the former summary should be updated accordingly to deliver most recent, authoritative, and correct information. For example, when the number of injuries increases in an Earthquake, the old number in the former summary must be replaced to avoid inconsistency of the summary. In this contribution we present a realtime event summarization system with explicit inconsistency detection. We model the realtime summarization problem as integer programming problems and solve the relaxed linear programming form by simplex method. To reduce the storage and computation cost of expensive inconsistency detection, we embed a fast inconsistency detection strategy in the simplex algorithm. Experiments on real twitter sets demonstrate the efficiency and effectiveness of our method.

1 Introduction

Accidents, disasters, political rallies...we are eager to gather information about different kinds of live events that happen around us. In the past, we rely on experienced journalists to cover the stories. At now, thanks to the large community of micro-blogging users, we are provided with instant reports published by individuals and organizations all over the world. More and more people today rely on microblogging contents, such as Tweets, to seek information about live events. However, the huge volume of event related tweets could be overwhelming. For example, TweeStudy shows that the majority (over 85%) of trending topics in microblog sphere are headline news and real-life events [5]. An event summarization system is needed to facilitate knowledge management and improve user experiences.

We have witnessed rapidly increasing popularity of research efforts in event summarization from tweets [16, 6, 13, 15, 7, 3, 20, 14]. Most previous work are based on extractive method, i.e. they extract a smallest set of representative tweet reports to form a brief summary. Extractive methods are easy to implement and have shown to perform well [16, 6, 13, 15, 7, 3, 20]. Our arguments are founded on extractive summarization methods.

Beyond the usual requirements for text summarization systems, such as coverage and representativeness of the summary, event summary must also be **realtime**. On one hand, the response must be fast. The summary must be efficiently updated as new tweets arrive. On the other hand, the summary must report the current status of the event. As an ongoing event often involves changing information, report must be updated to include new information when it emerges.

During the update process, the integrity of the summary must be preserved. An outdated report must be replaced if it leads to **inconsistency** in the summary. An inconsistent summary is harmful for most live events because it is confusing and misleading. In particular, for natural disaster or group incidents, users are interested on information such as number of injuries, suspects description and so on. We here list three scenarios when a former report needs to be replaced because of inconsistency. (1) The information changes as a natural consequence of event evolution. For example, in an earthquake the number of injuries is increasing over time. Thus the numbers in previous summaries become obsolete and they should be replaced by the most up-to-date numbers. (2) Multiple information sources provide conflicting information. For example, the number of injuries is often estimated by several parties, such as bystanders, hospitals and so on. When a more authoritative source, such as the local government announces the new estimate, the old estimates in previous summaries are no longer credible and must be replaced. (3) The information in previous summaries is wrong. For example, the police have suspected the wrong person and now they update the description. In this case, the realtime event summarization system must select the correct tweet to replace former reports.

Realtime event summarization from Tweets is still an open problem. In the literature, most of previous works treat the problem as producing different forms of summaries from a static set of tweets [16, 6, 13, 7, 3]. A few recent research works focused on efficient algorithms to summarize the tweet streams [15, 20]. Their summarization systems are based on coarse grained semantic analysis, and thus are not able to detect inconsistency. Though we have shown that integrity of the event summary is crucial, to the best of our knowledge, none of the previous works is able to produce a realtime event summarization which is guaranteed to exclude inconsistent information.

Two challenges arise in producing realtime event summarization without inconsistent information.

The first challenge lies in the macro-level algorithm. Realtime summarization requires an efficient algorithm to analyze the streaming tweets. As the amount of available tweets constantly increases to infinity, re-computation based on a complete set of all tweets up to the current timestamp is infeasible. The ideal algorithm is to incorporate new tweets as they become available, and discard old tweets when possible to limit the storage and speed up processing.

The second challenge is related to the micro-level analysis to detect inconsistency. Inconsistency detection is based on pair-wise similarity. Coarse grained semantic analysis, such as the cosine similarity measure based on the bag of words representation in previous works [16, 6, 13, 15, 7, 3, 20, 14] is suitable to capture topic similarity in a summarization, but is not able to detect inconsistent information. Inconsistency is revealed via word order and syntactic structures. We need to assess information similarity based on the combination of semantic, lexical and syntactic analysis. Furthermore, inconsistency detection is computationally expensive. It is important to avoid unnecessary pair-wise comparisons.

Our goal in this paper is to design a system that delivers realtime summary with integrity from tweets. To address the first challenge, we assume that, the realtime summarization problem given a small batch of new tweets can be modeled as two integer programming problems, one of which on the old tweets, and another on the new batch. Both integer programming problems can be relaxed to linear programming problems and be solved by the simplex method. In each update we first optimize the problem on the new batch. We use the solution on the new batch to modify the problem on the old tweets and incrementally update the summary. In this manner, we do not need to store or operate on the complete tweet set and the full similarity matrix.

To address the second challenge, we propose skeleton similarity: a new similarity metric to assess information similarity between any pair of tweets. An inconsistency detection strategy, which is a combination of the skeleton similarity and authority estimation heuristics, is then adopted in the pivoting operation in the simplex method. It has two advantages in embedding the skeleton similarity computation in the simplex algorithm. (1) It significantly reduces the number of information similarity comparisons. (2) It ensures that the former summary will be replaced by most up-to-date, authoritative and correct information.

Our contributions are three folds. (1) The integrity of event summary is a relatively unexplored area in Tweet summarization. We propose to improve the integrity of event summarization by explicit inconsistency detection. (2) Our system is targeted towards text streams. We model the realtime summarization problem as integer programming problems in small batches. We differ from existing work in that we enable incremental update in the simplex method framework. (3) We propose a novel skeleton similarity to efficiently and effectively capture inconsistency.

This paper is organized as follows. We briefly survey the related work in Sec. 2. In Sec. 3, we first introduce the idea of modeling a summarization problem as an integer programming problem and the standard simplex procedure to solve the relaxed linear programming problem. In Sec. 4, we give the problem definition for realtime event summarization given a small batch of new tweets and the improved simplex solution. The inconsistency detection strategy is a component of the simplex update algorithm. We present and analyze the experimental results on a real data set in Sec. 5. We conclude our work and suggest future directions in Sec. 6.

2 Related Work

Tweet summarization belongs to the more general category of multi-document summarization. We start by reviewing the common approaches in multi-document summarization. Then we move on to describe the recent research trend of tweet summarization.

2.1 Multi-document Summarization

Multi-document summarization conveys the main and most important meaning of several documents. There are generally two types of summarization techniques. One type is extraction-based summarization, which extracts objects from the entire collection and combines the objects into a summary without modifying the objects themselves. The other type is abstraction-based summarization which rephrases the source document. The majority of summarization systems are extractive. The extracted objects are often sentences. The selection is usually based on the representativeness of sentences, i.e. with significant frequency [19], or is a structural centroid in a sentence graph [**?**,6], or is considered important by a submodularity function [**?**].

Recently, a number of studies devote to summarizing documents related to events, mostly news articles. In [11] a main theme is extracted by selecting representative sentences in each time segment of the event. ETS [17] returns the evolution skeleton along the timeline by extracting representative and discriminative sentences at each phase. In [18] representative sentences are chosen based on relevance, coverage, coherence and cross-date diversity.

2.2 Tweet Summarization

The emergence of Twitter motivates recent research works on summarizing microblogging contents. Tweet summarization systems are successfully applied in entity-centric opinion summarization [9], personal summarization of interesting content [10, 1], search results grouping [8], and summarizing tweets for natural or social events [16, 6, 13, 15, 7, 3, 20].

At the algorithm level, tweet summarization also use extractive and abstractive methods. Except a few works which are based on abstractive method [14], most tweet summarization methods rank and select the most representative tweets. A few recent works start to improve general multi-document summarization methods for better efficiency. In [15], an incremental clustering method is presented. In [20] the selection range is shrinked by detecting sub-events and selecting one sentence with maximal similarity to any new sub event.

To achieve a better performance, the noisy and social nature of microblogs must be taken into consideration. Most tweet summarization systems will identify influential tweets [4], promote most recent tweet [2], and circumnavigate spam and conversational posts [3]. However, the integrity of summary has not yet been fully studied. The work that is most related to ours is the classification and summarization of situational information in [13, 12]. However they do not explicitly identify inconsistency as they simply provide all versions of inconsistent information. And they do not accelerate algorithms for realtime response.

3 Static Summarization

The standard summarization task is conducted on a static set of tweets. We refer this task as a static summarization. In this section, we first model the static summarization problem as an integer programming problem. We then present a high level explanation about the simplex method. We finally give an outline for the algorithm to solve static summarization.

3.1 Problem Definition

Suppose that we have a universe of N *candidate* tweets, within which M tweets are credible and relevant. The credible and relevant tweets are important so they are considered to be the *seeds* of the summary. The extractive method for any static summarization is to select a few representative tweet *reports*[1] from the tweet universe to form the summary. To model this problem, we use a vector $\mathbf{x} \in \mathcal{R}^N$, where each element $\mathbf{x}_j \in \{0, 1\}$ is a binary variable. If a candidate i is chosen to be a report in the summary, the corresponding $x_i = 1$. Otherwise, we set $x_i = 0$. We use another N-dimensional vector $\mathbf{c} \in \mathcal{R}^M$ to describe the loss of choosing each candidate as a report. $\mathbf{A} \in \mathcal{R}^{M\times N}$ is a similarity matrix, where $A_{i,j}$ is the similarity between a seed i and a candidate j in the tweet universe. $\mathbf{b} \in \mathcal{R}^M$ is a weight vector, where $b_i > 0$ indicates the importance of seed i being covered in the summary. Our objective is to

$$\min \mathbf{c}^T\mathbf{x} \text{ subject to } \mathbf{Ax} \geq \mathbf{b}, \forall i \ x_i \in \{0, 1\}. \tag{1}$$

The optimization problem in Equ. 1 aims to find a minimal set of reports that deliver all credible and relevant information. Thus the result summary achieves brevity, coverage and representativeness. The decision of b, c and the M indices affects the performance of summarization. We allow the flexibility of choosing any sounding technique to determine the credible and relevant set of tweets, their importance value and the loss of choosing any candidate. In this work, we choose the top M results from a search engine, determine $\mathbf{b}$ by estimating the authority of each tweet account, and assign $\mathbf{c}$ by assessing the textual quality of each candidate. The technical details will be discussed in Sec. 5.

3.2 The Simplex Method

We transform the integer programming problem in Equ. 1 to a bounded linear programming problem by making the following adjustments: $\tilde{\mathbf{c}} = [\mathbf{c}, \mathbf{0}], \tilde{\mathbf{x}} = [\mathbf{x}, \mathbf{z}]^T, \tilde{\mathbf{A}} = [\mathbf{A}, -\mathbf{I}]$, where $\mathbf{z} \in \mathcal{R}^M$, $\mathbf{I}$ is the $M \times M$ identity matrix. Therefore we have the following objective

$$\min \tilde{\mathbf{c}}\tilde{\mathbf{x}} \text{ subject to } \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{b}, \forall 1 \leq j \leq N, \ \tilde{x}_j \leq 1, , \tilde{\mathbf{x}} \geq \mathbf{0}. \tag{2}$$

The linear programming problem in Equ. 2 can be solved by the simplex method. Each iterate in the simplex method is a *basic feasible point* that (1) it satisfies $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{b}, \forall 1 \leq i \leq M + N, \tilde{x}_i \geq 0, \forall 1 \leq j \leq N, \tilde{x}_j \leq 1$ and (2) there

[1] To distinguish the three types of tweets, we will refer the tweets to be summarized as candidates, the tweets in the summary as reports, and the credible and relevant tweets as seeds.

exists three subsets $\mathcal{B}$,$\mathcal{U}.\mathcal{L}$ of the index set $\mathcal{A} = \{1, 2, \cdots M + N\}$ such that $\mathcal{B}$ contains exactly M indices, $\mathcal{A} = \mathcal{B} \cup \mathcal{U} \cup \mathcal{L}$ and $i \in \mathcal{U} \Rightarrow \tilde{x}_i = 1, i \in \mathcal{L} \Rightarrow \tilde{x}_i = 0$.

The major issue at each simplex iteration is to decide which index to be removed from the basis $\mathcal{B}$ and replaced by another index outside the basis $\mathcal{B}$. As the optimal is achieved when the KKT conditions are satisfied, in each simplex iteration first a **pricing** step is conducted to check on the KKT conditions. If the KKT conditions are satisfied then the problem is solved. Otherwise a **pivoting** operation is implemented to select entering and leaving indices.

We should keep in mind that the simplex iterations are run on basic feasible points. To obtain an easy start of basic feasible points, we solve the following linear programming problem.

$$\min \mathbf{e}^{\mathbf{T}}\mathbf{s} \text{ subject to } \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \mathbf{I}\mathbf{s} = \mathbf{b}, \forall 1 \le j \le N,\ \tilde{x}_j \le 1, , \tilde{\mathbf{x}} \ge \mathbf{0}, \mathbf{s} \ge \mathbf{0}, \tag{3}$$

where $\mathbf{e}$ is the vector of all ones, $\mathbf{I}$ is a diagonal matrix whose diagonal elements are $I_{ij} = 1$. $\mathbf{s}$ are called artificial variables. It is easy to see that the solution to Equ. 3 is a basic feasible point for Equ. 2, if the objective $\mathbf{e}^{\mathbf{T}}\mathbf{s} = 0$.

The remaining problem is that the solution we obtained for Equ. 2 are not integers. As our interest is on the variables $\mathbf{x}$ in Equ. 1, we can intepretate the values obtained for Equ. 2 $\tilde{x}_j, \forall 1 \le j \le N$ as the probability of choosing j as a response. For indices that are in the upper bound set $j \in \mathcal{U}$, the corresponding candidate is deemed in the summary. For indices that are in the lower bound set $j \in \mathcal{L}$, the corresponding candidate is filtered. For the remaining candidates, we perform a **rounding** algorithm. We first sort the solution we obtained for Equ. 2 in descending order of its values $0 < \tilde{x}_j < 1, \forall 1 \le j \le N$ and then sample x_j based on the probability of assigned value $\tilde{x}_j$.

$$p(x_j = 1) = \tilde{x}_j, \forall 1 \le j \le N \tag{4}$$

To conclude this section, we present Algorithm. 1. The frameowk of static summarization includes three steps. The first stage (step 1 to step 2) is to obtain a basic feasible point for the linear programming problem Equ. 2, the second stage (step 3 to step 4) is to optimize the linear programing problem in Equ. 2, and the third stage (step 5) is to obtain the integer solutions for Equ. 1.

4 Dynamic Summarization

For streaming text, the realtime summarization system must be able to update the summary as new contents arrive. The summary is dynamic because it changes over time. We here consider a batch setting: the summary is updated when a predefined number T of tweets are received. Note that if we set $T = 1$ we enable online update. In this section, we first model the dynamic summarization problem and then we present the improved simplex method.

```
Input: $\mathbf{c}, \mathbf{b}, A$
Output: $\mathbf{x}$
Initialize $\tilde{\mathbf{x}} = \mathbf{0}, \tilde{\mathbf{A}} = [\mathbf{A}, -\mathbf{I}], \mathbf{s} = \mathbf{b}$;
Solve $\min \mathbf{e^T s}$ subject to $\tilde{\mathbf{A}}\tilde{\mathbf{x}} + \mathbf{Is} = \mathbf{b}, \forall 1 \leq j \leq N,\ \tilde{x}_j \leq 1, , \tilde{\mathbf{x}} \geq \mathbf{0}, \mathbf{s} \geq \mathbf{0}$ by
  simplex method;
Initialize $\tilde{\mathbf{x}}$ by the solution above, set $\tilde{\mathbf{c}} = [\mathbf{c}, \mathbf{0}]$ ;
Solve $\min \tilde{\mathbf{c}}\tilde{\mathbf{x}}$ subject to $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \mathbf{b}, \forall 1 \leq j \leq N,\ \tilde{x}_j \leq 1, , \tilde{\mathbf{x}} \geq \mathbf{0}$ by simplex
  method;
$\tilde{\mathbf{x}} = [\mathbf{x}, \mathbf{z}]^T$, Rounding $\mathbf{x}$;
```

Algorithm 1: The framework for static summarization

4.1 Problem Definition

When a small batch of tweets arrive, the candidate universe will be a new set of N_1 candidates with importance weights denoted by $\mathbf{c_1}$ and the former candidates. We only keep previous reports $\mathbf{x_0}$ with $x_i = 1$ in the former summary and their loss value $\mathbf{c_0}$ and discard everything else from former corpus to reduce storage space.

To produce a summary that covers the demanded information, the model in Sec. 3 requires a predefined set of credible and relevant seed tweets and their importance weights $\mathbf{b}$. It is crucial here to mention that in the dynamic setting the seed set is priorly unknown due to inconsistency. For example, if a new credible tweet is found to conflict with a former credible tweet, then the former one must be excluded. Suppose we have M seeds, but the integrity is not guaranteed. It is possible to check on inconsistency to prune the M seeds to identify a consistent new set. However, the inconsistency detection algorithm on $M \times (M-1)$ pairs is expensive. Henceforth, our idea is to call the inconsistency detection algorithm only when it is necessary.

Suppose we segment the timeline of tweets into two divisions. M_0 represents the set of seeds that are covered by the former summary, M_1 is the set of seeds published after the former summary was generated. M_1 is generated as follows. We first retrieve R relevant tweets published after the summary. Then we can adopt any heuristic to fast prune inconsistent seeds in M_1. We can also adopt any external strategy to exclude seeds that are less credible than any seed in M_0. Therefore the seeds in M_1 must be covered with priority if they are inconsistent with the seeds in M_0, because any seed in M_0 is newer and at least as credible as any seed in M_1. To deliver an up-to-date summary, we set $\mathbf{c}$ so that the importance for new seeds is larger than the old seeds.

Below we give the architecture of the dynamic summarization system. We model the dynamic summarization as sequential integer programing problems. Given the whole set of candidates $\mathbf{x_0}, \mathbf{x_n}$ and the associated loss $\mathbf{c_0}, \mathbf{c_n}$, the importance of possible seeds $\mathbf{b_1}$, at stage I, we first solve the optimization problem for the higher division of seeds.

$$\min \mathbf{c_0}^T\mathbf{x_0}+\mathbf{c_1}^T\mathbf{x_1} \text{ subject to } S\mathbf{x_0}+B\mathbf{x_1} \geq \mathbf{b_1}, \forall i, \forall j \in \{0,1\}\ x_{j,i} \in \{0,1\}. \quad (5)$$

In solving Equ. 5 we will obtain the refined seed set $\mathbf{b_0}$. In stage II we solve the optimization problem:

$$\begin{gathered} \min \mathbf{c_0}^T\mathbf{x_0} + \mathbf{c_1}^T\mathbf{x_1} \\ s.t. \begin{bmatrix} F & D \\ S & B \end{bmatrix} \begin{bmatrix} \mathbf{x_0} \\ \mathbf{x_1} \end{bmatrix} \geq \begin{bmatrix} \mathbf{b_0} \\ \mathbf{b_1} \end{bmatrix}, \\ \forall i, \forall j \in \{0,1\}\ x_{j,i} \in \{0,1\}, \end{gathered} \quad (6)$$

where $F \in \mathcal{R}^{M_0 \times N_0}$ is the similarity matrix between former reports and old tweets, $D \in \mathcal{R}^{M_0 \times N_1}$ is the similarity matrix between new candidates and old credible tweets, $S \in \mathcal{R}^{M_1 \times N_0}$ is the similarity matrix between old reports and new credible tweets, $B \in \mathcal{R}^{M_1 \times N_1}$ is the similarity matrix between new candidates and new credible tweets.

4.2 Simplex Update Method

As in Sec. 3, we introduce new variables. $\tilde{\mathbf{c}} = [\mathbf{c_0}, \mathbf{c_1}, \mathbf{0}], \tilde{\mathbf{x}} = [\mathbf{x_0}, \mathbf{x_1}, \mathbf{z}]^T, \tilde{\mathbf{A}} = [F, B, -\mathbf{I}], \mathbf{s}$, where $\mathbf{z} \in \mathcal{R}^{M_1}$, $\mathbf{I}$ is the $M_1 \times M_1$ identity matrix. To initialize, we set $\mathbf{x_0} = \mathbf{1}$ and keep $\mathbf{x_0}$ fixed while solving Equ. 3. We use the solution as the initialized guess for the second stage optimization.

In each iterate of simplex, we first operate the pricing step. Let's compute $\bar{c_j} = c_j - y^T\mathbf{A}'_{,j}$ for every $\tilde{x_j}$ not in the basis, where $\mathbf{y} = \mathbf{B}^{T^{-1}}\mathbf{c_B}$,. If $\tilde{x_j} = 0, \bar{c_j} > 0$ and $\tilde{x_j} = 1, \bar{c_j} < 0$, then the objective function is optimized. Otherwise we implement the pivoting operation.

We modify the conventional pivoting operation in the simplex method. In the initialization we keep $\mathbf{x_0} = \mathbf{1}$, which indicates that the constraint is satisfied with all the former reports kept. If the pricing step suggests the objective function is not optimized, there are usually multiple indices which violate the KKT conditions. Hence in selecting the entering index, we prefer a z index, because selecting an auxiliary variable leads to minimal modification of the summary. However, if an index from $\mathbf{x_0}$ is to leave and an index from $\mathbf{x_1}$ is to enter, it implies that a new tweet is able to cover the same set of seeds. We can not rule out the possibility of inconsistency. Therefore we conduct inconsistency test between the entering index and the leaving index. If the leaving index is found to be inconsistent of the new report, we delete the corresponding index $x_{0,i}$ from iteration. The pivoting operation is described in Algorithm. **??**.

4.3 Inconsistency Detection

5 Experiment

5.1 Experimental Setup

We use the twitter event data set []. The corpus includes data for 30 different Twitter datasets associated with real world events. The datasets were collected

```
$q = \arg\max_j \{\bar{c_j} \forall j \in \{\mathbf{z_N}, \mathbf{s_N}, \mathbf{x_L}\}, -\bar{c_j} \forall j \in \mathbf{x_U}\};$
$d = \mathbf{B}^{-1}\mathbf{A}'_{,q};$
if $q \in \{\mathbf{z_N}, \mathbf{s_N}\}$ then
    $x_q^{new} = \min_i \left\{ \{\frac{x_{\mathbf{B}_i}}{d_i}, \frac{s_{\mathbf{B}_i}}{d_i}, \frac{z_{\mathbf{B}_i}}{d_i}\} \forall d_i > 0, \{\frac{x_{\mathbf{B}_i}-1}{d_i}\} \forall d_i < 0 \right\};$
    $\mathbf{x}'^{new}_{\mathbf{B}^{old}} = \mathbf{x}'^{old}_{\mathbf{B}^{old}} - dx_q^{new};$
    $p = \arg\min_i;$
    $\mathbf{B}^{new} \leftarrow \mathbf{B}^{old} - \{p\} \bigcup \{q\};$
end
if $q \in L$ then
    $x_q^{new} = \min_i \left\{ \{\frac{x_{\mathbf{B}_i}}{d_i}, \frac{s_{\mathbf{B}_i}}{d_i}, \frac{z_{\mathbf{B}_i}}{d_i}\} \forall d_i > 0, \{\frac{x_{\mathbf{B}_i}-1}{d_i}\} \forall d_i < 0, 1 \right\};$
    $\mathbf{x}'^{new}_{\mathbf{B}^{old}} = \mathbf{x}'^{old}_{\mathbf{B}^{old}} - dx_q^{new};$
    if $x_q^{new} \neq 1$ then
        $p = \arg\min_i;$
        $\mathbf{B}^{new} \leftarrow \mathbf{B}^{old} - \{p\} \bigcup \{q\};$
    end
end
if $q \in U$ then
    $x_q^{new} = 1 - \min_i \left\{ \{\frac{x_{\mathbf{B}_i}}{-d_i}, \frac{s_{\mathbf{B}_i}}{-d_i}, \frac{z_{\mathbf{B}_i}}{-d_i}\} \forall d_i < 0, \{\frac{1-x_{\mathbf{B}_i}}{d_i}\} \forall d_i > 0, 1 \right\};$
    $\mathbf{x}'^{new}_{\mathbf{B}^{old}} = \mathbf{x}'^{old}_{\mathbf{B}^{old}} + d(1 - x_q^{new});$
    if $x_q^{new} \neq 0$ then
        $p = \arg\min_i;$
        $\mathbf{B}^{new} \leftarrow \mathbf{B}^{old} - \{p\} \bigcup \{q\};$
    end
end
```

Algorithm 2: Pivoting in the improved simplex method

between 2012 and 2016, always using the streaming API with a set of keywords. Different types of tweets are presented, including repliesand retweets. The corpus is multilingual, including English, Japanese and so on. For the purpose of event summarization, we select 10 events and filter relevant tweets by only select tweets in English. More details of the data set are illustrated in Table 5.1.

Table 1. Statistics of the data set

Event	Number of tweets	Time period	Average Length of tweets	Number of terms
EOutbreak	12983	2014/7/1 ∼ 2014/7/31	93	15
GUattack	21000	2014/6/2 ∼ 2014/7/17	103	15
HProtest	27000	2014/9/26 ∼ 2014/10/17	96	14
THagupit	10315	2014/12/5 ∼ 2014/12/11	92	14
CHShoot	15000	2015/1/7	95	15
HPatricia	9288	2015/10/24 ∼ 2015/12/8	88	13
RWelcome	19725	2015/9/2 ∼ 2015/11/24	103	15
BAExplossion	15000	2016/3/22	91	14
HPCyprus	14917	2016/3/29 ∼ 2016/3/30	92	15
LBlast	13423	2016/3/27 ∼ 2016/3/30	95	15

In pre-processing corpus, we use the Bloom Filter algorithm to filter duplicate tweets. Emoji expressions,http links and mentions (@somebody)are removed from the vocabulary.

Ground truth For the 10 events in corpus,2 students are invited to manually extract tweets to form the summarization for each event.

The comparative methods are state-of-the-art summarization methods.

(1)LPR

(2)MSSF

(3)SNMF

(4)Sumblr

We implemented all algorithms in java and all experiments have been executed on a server with an AMD Opteron Processor 280 2.40G Hz (4 cores)and main memory 8G bytes.

5.2 Summarization Performance

The measurement is mainly based on Recall-Oriented Understudy for Gisting Evaluation(ROUGE) an evaluation toolkit for document summarization[23] which automatically determines the quality of a summary by comparing it with the human generated summaries through counting the number of their overlapping textual units (e.g., n-gram, word sequences, and etc.). In particular, F-measure scores of 8 evaluations are presented for our experiments.

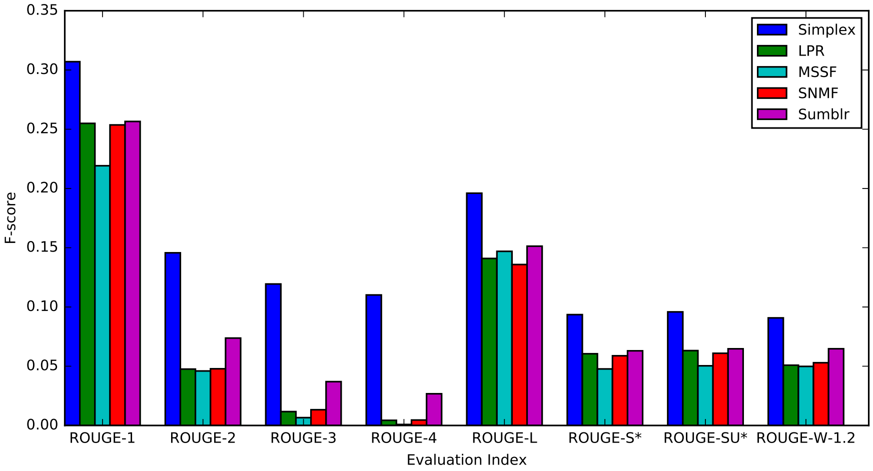


Table 2. the updateRatio of Simplex

Event	Time1	Time2	Time3
EOutbreak	0	0.57	0.38
GUattack	0	0.36	0.33
HProtest	0.38	0.67	0.96
THagupit	0	0.1	0.62
CHShoot	0.94	0.62	0.25
HPatricia	0	0.75	0.83
RWelcome	0.20	0.50	0
BAExplossion	1	0.67	0.80
HPCyprus	0.88	0.65	0.76
LBlast	0.22	0.50	0.69

Table 3. the updateRatio of Sumblr

Event	Time1	Time2	Time3
EOutbreak	0.75	0.30	0.20
GUattack	0.65	0.50	0.30
HProtest	0.35	0.40	0.30
THagupit	0.65	0.45	0.10
CHShoot	0.20	0.05	0.10
HPatricia	0.25	0.20	0
RWelcome	0.35	0.35	0.20
BAExplossion	0.45	0.35	0
HPCyprus	0.30	0.15	0.25
LBlast	0.30	0.25	0.05

Table 4. the updateRatio of MSSF

Event	Time1	Time2	Time3
EOutbreak	0.41	0.29	0.24
GUattack	0.62	0.18	0.18
HProtest	0.65	0.22	0.17
THagupit	0.39	0.61	0.18
CHShoot	0.33	0.39	0.18
HPatricia	0.47	0.29	0
RWelcome	0.44	0.62	0.17
BAExplossion	0.39	0.12	0.12
HPCyprus	0.11	0.17	0.17
LBlast	0.35	0.50	0.33

5.3 Inconsistency Detection Performance

In addition of the state-of-the-art summarization methods, for comparison on inconsistency detection, we adopted some naive methods

Table 5. the inconsistency of Sumblr

Event	Time0	Time1	Time2	Time3
EOutbreak	0.65	0.70	0.65	0.70
GUattack	0.60	0.55	0.40	0.50
HProtest	0.60	0.55	0.55	0.50
THagupit	0.70	0.75	0.70	0.65
CHShoot	0.85	0.70	0.75	0.70
HPatricia	0.75	0.75	0.75	0.75
RWelcome	0.65	0.60	0.65	0.65
BAExplossion	0.80	0.60	0.60	0.55
HPCyprus	0.50	0.55	0.55	0.55
LBlast	0.65	0.60	0.55	0.55

Table 6. the inconsistency of MSSF

Event	Time0	Time1	Time2	Time3
EOutbreak	0.76	0.71	0.76	0.78
GUattack	0.62	0.59	0.59	0.47
HProtest	0.27	0.33	0.44	0.44
THagupit	0.83	0.83	0.82	0.82
CHShoot	0.83	0.78	0.76	0.71
HPatricia	0.71	0.71	0.76	0.76
RWelcome	0.69	0.75	0.72	0.72
BAExplossion	0.67	0.71	0.76	0.83
HPCyprus	0.83	0.89	0.72	0.65
LBlast	0.41	0.56	0.50	0.56

5.4 Efficiency Study

5.5 Effects of Parameters

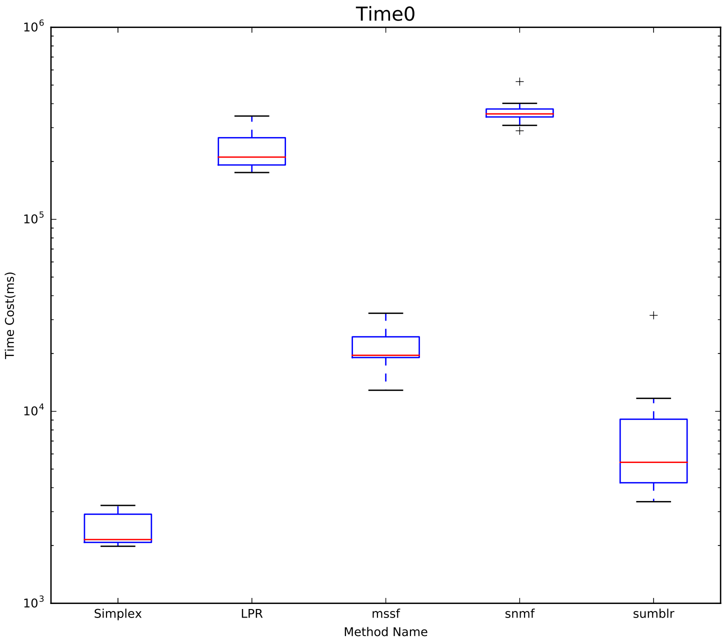
6 Conclusion

References

1. J. Y. Chin, S. S. Bhowmick, and A. Jatowt. Totem: Personal tweets summarization on mobile devices. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pages 1305–1308, New York, NY, USA, 2017. ACM.

Table 7. the inconsistency of others

Event	SNMF	LPR
EOutbreak	0.67	0.80
GUattack	0.47	0.40
HProtest	0.20	0.10
THagupit	0.33	0.20
CHShoot	0.60	0.60
HPatricia	0.53	0.50
RWelcome	0.33	0.20
BAExplossion	0.33	0.30
HPCyprus	0.33	0.30
LBlast	0.07	0.10



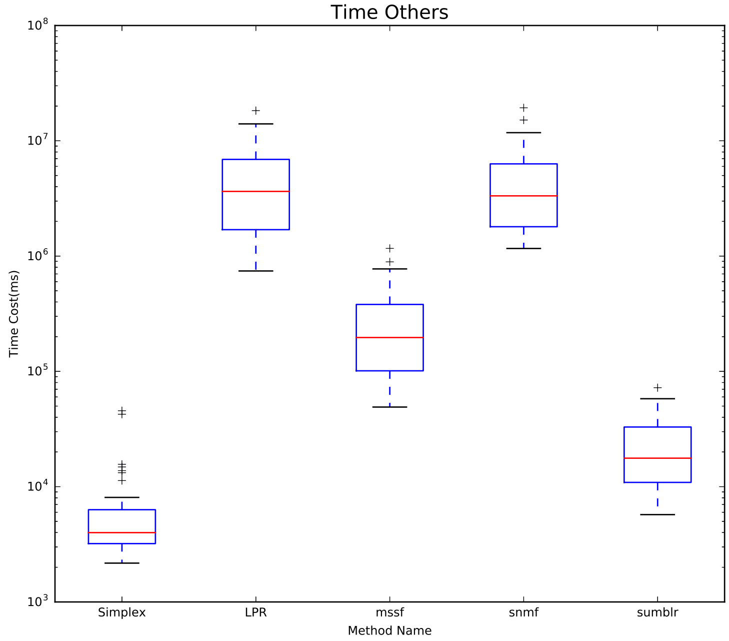
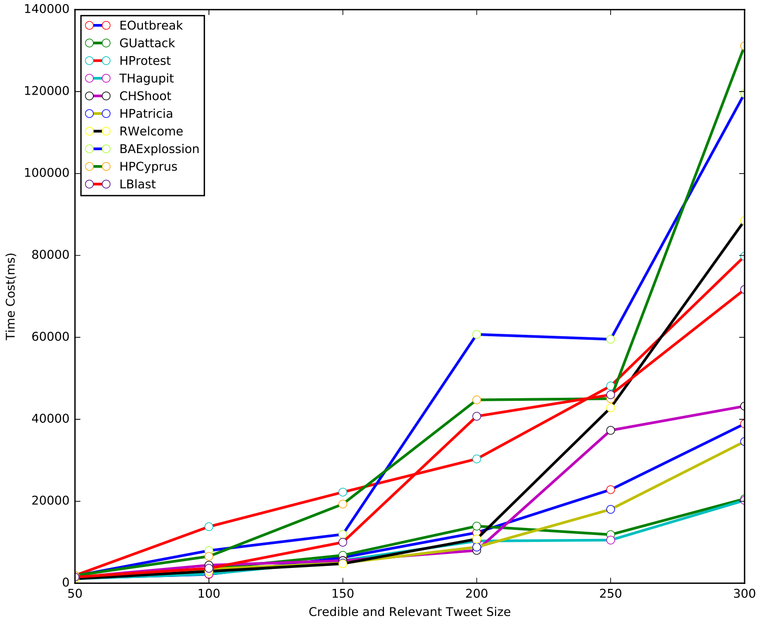
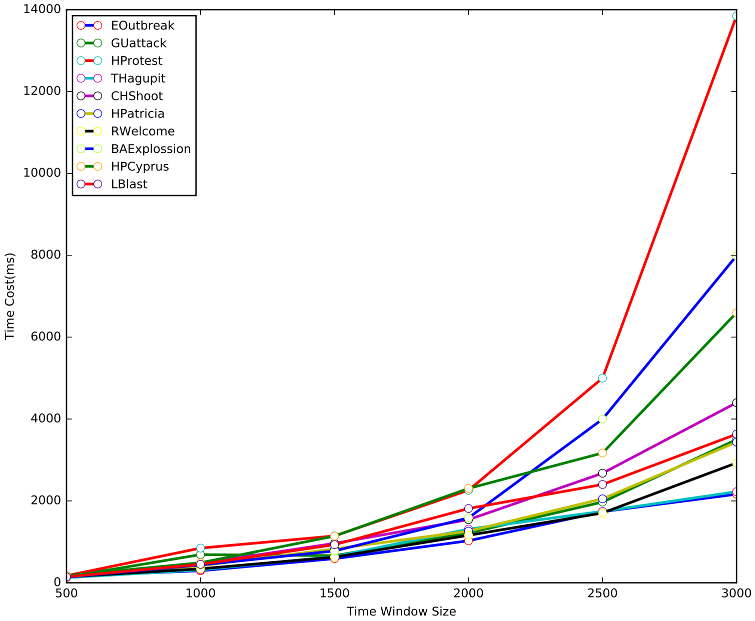

Time Others
10^8
10^7
10^6
10^5
10^4
10^3
Time Cost(ms)
Simplex
LPR
mssf
snmf
sumblr
Method Name

Fig. 1. fig2


EOutbreak
GUattack
HProtest
THagupit
CHShoot
HPatricia
RWelcome
BAExplossion
HPCyprus
LBlast
140000
120000
100000
80000
60000
40000
20000
0
Time Cost(ms)
50
100
150
200
250
300
Credible and Relevant Tweet Size



2. M. Efron and G. Golovchinsky. Estimation methods for ranking recent information. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, SIGIR '11, pages 495–504, New York, NY, USA, 2011. ACM.

3. M. Gillani, M. U. Ilyas, S. Saleh, J. S. Alowibdi, N. Aljohani, and F. S. Alotaibi. Post summarization of microblogs of sporting events. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 59–68, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

4. J. Hannon, M. Bennett, and B. Smyth. Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 199–206, New York, NY, USA, 2010. ACM.

5. H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM, 2010.

6. C. Lin, C. Lin, J. Li, D. Wang, Y. Chen, and T. Li. Generating event storylines from microblogs. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 175–184, New York, NY, USA, 2012. ACM.

7. Z. Liu, Y. Huang, and J. R. Trampier. Leds: Local event discovery and summarization from tweets. In *Proceedings of the 24th ACM SIGSPATIAL International*

Conference on Advances in Geographic Information Systems, GIS '16, pages 53:1–53:4, New York, NY, USA, 2016. ACM.

8. M. Mathioudakis and N. Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 international conference on Management of data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM.
9. X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 379–387, New York, NY, USA, 2012. ACM.
10. Z. Ren, S. Liang, E. Meij, and M. de Rijke. Personalized time-aware tweets summarization. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 513–522, New York, NY, USA, 2013. ACM.
11. F. ren Lin and C.-H. Liang. Storyline-based summarization for news topic retrospection. *Decision Support Systems*, 45(3):473 – 490, 2008. ¡ce:title¿Special Issue Clusters¡/ce:title¿.
12. K. Rudra, S. Banerjee, N. Ganguly, P. Goyal, M. Imran, and P. Mitra. Summarizing situational tweets in crisis scenario. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, HT '16, pages 137–147, New York, NY, USA, 2016. ACM.
13. K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh. Extracting situational information from microblogs during disaster events: A classification-summarization approach. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 583–592, New York, NY, USA, 2015. ACM.
14. B. Sharifi, M.-A. Hutton, and J. Kalita. Summarizing microblogs automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 685–688, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
15. L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: continuous summarization of evolving tweet streams. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '13, pages 533–542, New York, NY, USA, 2013. ACM.
16. H. Takamura, H. Yokono, and M. Okumura. Summarizing a document stream. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 177–188, Berlin, Heidelberg, 2011. Springer-Verlag.
17. D. Wang, L. Zheng, T. Li, and Y. Deng. Evolutionary document summarization for disaster management. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 680–681. ACM, 2009.
18. R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pages 745–754. ACM, 2011.
19. W. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI*, volume 7, 2007.
20. A. Zubiaga, D. Spina, E. Amigó, and J. Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, HT '12, pages 319–320, New York, NY, USA, 2012. ACM.