Exploiting Social Networks as a Live Mass Media Channel During Disasters for Reactions

Minh-Tien Nguyen[a,b], Tri-Thanh Nguyen[d], Asanobu Kitamoto[c], Minh-Le Nguyen[a,*]

[a] *School of Information Science,*
Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan.
[b] *Hung Yen University of Technology and Education (UTEHY), Hung Yen, Vietnam.*
[c] *National Institute of Informatics (NII),*
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.
[d] *Vietnam National University, Hanoi (VNU),*
University of Engineering and Technology (UET),
144, Xuan Thuy, Cau Giay, Hanoi, Vietnam.

Abstract

Social networks, e.g. Twitter have been proved to be an almost real-time means of spreading information and can be exploited as a valuable information channel including emergencies such as disasters, during which people need updated information for reasonable reactions. This paper presents a framework designed to distill informative information in a form of actionable tweets of casualties, cautions, and donations for providing users live information for quick responses during a disaster. The framework has to tackle tweet challenges such as diversity, large volume, and noise by utilizing several techniques: a) retrieves a large number of tweets for a good coverage to ensure the diversity; b) removes irrelevant and indirect tweets or noise for reducing the volume; c) divides informative tweets into valuable-predefined classes for quick navigation, and groups them in a class into a number of topics to preserve the diversity; and finally, d) ranks

☆This manuscript is an improved and extended version of the paper: TSum4act: A Framework for Retrieving and Summarizing Actionable Tweets during a Disaster for Reaction, presented at the 19^{th} Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) 2015, Ho Chi Minh City, Vietnam.

*Corresponding author

Email addresses: `tiennm@jaist.ac.jp` (Minh-Tien Nguyen), `ntthanh@vnu.edu.vn` (Tri-Thanh Nguyen), `kitamoto@nii.ac.jp` (Asanobu Kitamoto), `nguyenml@jaist.ac.jp` (Minh-Le Nguyen)

the tweets to summarize the topics to be compact for user's quick scan. In the ranking step, we propose to exploit event extraction for enriching the semantics and reducing the noise of tweets. To validate the efficiency of our framework, we take Twitter as a case study. Experimental results on 230,535 tweets collected during the Joplin tornado indicate that by incorporating the event extraction, our method significantly outperforms 3.24%, 13.1% and 16.86% of completeness over no-ranking, no-extraction and retweet baseline.
Keywords: Disaster Reactions, Crisis Responses, Event Extraction, Tweet Summarization.

The Notes of Extended Manuscript

The notes:

- Paragraphs with the first sentence highlighted by **Apricot color** are added or extended by adding new information compared to the original version.
- Sections with New section are newly added compared to the original version.

The extensions: Compared to the original paper, this manuscript makes five new and significant improvements as follows.

- It investigates a literature review, which makes a story of tweet summarization in common as well as disaster scenarios.
- It refines and clearly describes our model, which are not sufficiently mentioned in the original paper. This manuscript is significantly improved compared to the original version.
- It also compares our method to two additional methods: no event extraction, which does not use the event extraction and no ranking, which randomly selects tweets as the summarization. Experimental results indicate that our method significantly outperforms the two additional baselines as well as the retweet model.
- It observes the output from our model to analyze its advantages and limitations in extracting tweets.
- We carefully revise the English usage, e.g. typos, grammar in the manuscript.

1. Introduction

Thanks to the fast growth of Internet connected devices, e.g. computers, smart phones and social networks (e.g. Twitter with a large number of users) become ubiquitously accessible and a live mass media channel for spreading information including natural or man-made disasters (Sakaki et al., 2010; Vieweg et al., 2010; Nguyen et al., 2015; Yates and Paquette 2011; Anderson 2012; Verma et al., 2011). Let's take Twitter as an example. Social users spread information in the form of tweets, a short text message with the maximal length of 140 letters, mentioning a wide range information in all aspects of life including disaster facts. (Vieweg 2012) analyzed tweets during four natural disasters and pointed out that tweets during disasters can be classified into 32 categories contributing valuable information to users' decision making in such emergencies. Such beneficial need inspires an interesting idea that Twitter can be utilized as an informative information channel in dealing disasters with two advantages. Firstly, it is a real-time system, viz. once a tweet of a user is posted, all of his/her followed users can read it immediately. Secondly, Twitter has a large number of user communities, in which each can be regarded as a volunteer/sensor, can collect live information in a large affected area, thus, providing users all updated aspects of disasters (Sakaki et al., 2010).

During a disaster, tweets usually tend to explode in a large volume and high speed challenging people in capturing valuable information for making suitable reactions. To tackle this issue, Twitter provides a search function, which returns a large number of irrelevant tweets ranked by retweet number in a reverse chronological order. It obviously makes a difficulty for users to read and find out their interests. One possible solution to address the redundance is tweet summarization (O'Connor et al., 2010; Chakrabarti and Punera 2011; Ritter et al., 2012; Ma et al., 2012; Khan et al., 2013; Wu et al., 2013; Rudra et al., 2015). These approaches usually base on lexical or surface representation such as term frequency – inverted document frequency (TF-IDF), which face the noise of tweets, e.g. incorrect lexical presentation and special characters such as text based emotions. Also, they may not exploit adherent semantics of tweets such as entities in term of times, locations, numbers, which plays an important role in tweet summarization (Xu et al., 2013).

The objective of our study is to automatically extract informative tweets from a large tweet set posted during disasters for user reactions. The key insight behind our framework is that it exploits event extraction to increase semantics and reduce the noise of tweets in building an event graph for ranking. More precisely, given a disaster-related query, our framework: 1) retrieves tweets containing the query from the source; 2) obtains informative tweets in valuable classes by removing irrelevant ones; 3) assigns the informative tweets into clusters; and finally, 4) ranks to select top m tweets in each cluster to provide for users as the summarization. The pipeline model allows to tackle information challenges in disasters such as large tweet volume, noise, and diversity, i.e. several aspects and topics. This paper makes the following contributions:

- It filters data to get relevant tweets to deal with the noise in term of unrelated tweets.
- To preserve the diversity of tweets in the form of several topics, it divides tweets into valuable classes. Then, each class is further divided into a variable number of clusters, each of which corresponds to a topic.
- It adapts extractive summarization for each cluster to obtain a small tweet set as the recommendation to the users, in order to deal with the large volume. Near duplicated removal is also applied to make better results.
- It proposes to present tweets in the form of events consisted of *subject*, *action*, *location*, and affected *number* (of damages) to solve the lexical noise (e.g., spelling errors in informal communication, different expressions of the same fact), and to exploit the semantics of the tweets. The above information helps to answer common important questions related to a fact of the disaster, e.g., *what*, *where*, and *how many*?

- It successfully applies our framework[1] to a real dataset collected during the Joplin tornado. Experimental results indicate that our method significantly outperforms 3.24%, 13.1% and 16.86% of completeness over three baselines: no-ranking, no-extraction and retweet, correspondingly. Promising results suggest that information from our framework can be combined with other sources, i.e. TV, online news, or emergency services to provide meaningful information for people during disasters.

This paper is organized as the following. We next overview related work in §2. The our proposed framework is shown in §3, along with data preparation and the process of the framework. After extracting informative tweets, we present baselines and evaluation method in §4. Subsequently, we report summary performance and results of other components in our framework with discussion in §5. We finish by drawing conclusions in §6

2. Related Work

This section overviews related work of our study. We first show research in common tweet summarization and next introduce the summarization in the literature in the context of crisis response.

2.1. Tweet Summarization

New section

Tweet summarization makes a new direction in extracting salient messages collected from Twitter. (Ritter et al., 2012) introduced a method which automatically extracts open domain events. The authors define an event as a frame including information slots such as time, entities, and event phrases. Events are ranked and classified before plotting on an event calender. The result increases 0.14 of F1 over the method without using NER. (O'Connor et al., 2010) presented a new search model, which groups tweets by their significant terms. This model facilitates navigation and drilldown via a faceted search interface.

[1]The output of system can be seen at: http://150.65.242.101:9294

(Chakrabarti and Punera 2011) proposed a model for summarizing information in football matches, which had already been detected. Based on this assumption, the authors use Hidden Markov Model to identify sub-events of a parent event in time segment with 0.5 of precision and 0.52 of recall. To summarize information, the authors exploit TF-IDF combining with Cosine similarity. (Khan et al., 2013) summarized tweets in a debating event by using lexical level underlying topical modeling and graphical model. Summary performance is around 0.816 of precision and 0.80 recall on their datasets.

(Wang et al., 2015) presented *Sumblr*, which tackles the continuous summarization aspect from tweet streams. Sumblr contains three modules: tweet clustering, online and historical summarization, and topic detection. Experimental results on large-scale tweets demonstrate the efficiency and effectiveness of this approach. (Yajuan et al., 2012) exploited social influence from users and content quality to rank tweets for topic summarization. This models first segments a tweet stream into sub-topics and then computes the tweet score by integrating user information and tweet content. The high-quality summaries are decided by a classifier with several refined features. The model obtains 0.4167 in ROUGE-1 over an earthquake dataset.

2.2. Tweet Summarization in Disaster

New section

The growth of Twitter provides a new method for spreading information in disasters. (Sakaki et al., 2010) extracted earthquake information and spreaded them to users by using social sensors. Results show that their system can convey earthquake information faster than government information channel about 3-4 seconds. In another research, (Imran et al., 2013) focused on extracting nugget information from a tornado. The authors present a model including two important modules: classification and extraction, in which the AUC is 0.79 and Hit ratio is 0.983, correspondingly. The extraction could be denoted as summarization, in which the snippets of information are extracted and given for users. (Rudra et al., 2015) proposed a model for extracting situational information from microblogs during disasters by classification approach. The model is presented in two steps: (1) extracting the situational information from a large of sentiment and opinion tweets and (2) summarizing the situational tweets. For extraction, Support Vector Machines (SVMs) with a set of features are used. For summarization, the authors use content-word-based summarization in form of Integer Linear Programming (ILP) with a set of constrains. This model obtains improvements over baselines in four disaster datasets. (Rudra et al., 2016) extended their former work (Rudra et al., 2015) by presenting a model, which includes two steps: extraction and abstraction. In the first step, important tweets are selected for the second step, which employs ILP with a set of constraints to generate the final outputs. This model obtains promising results in disaster scenarios.

The system in (Imran et al., 2013) is perhaps the most relevant to our framework. It extracts nugget information collected during a tornado for crises response by using classification and extraction approaches. In the classification, it uses annotated data with pre-defined features to select informative tweets, which are used for the extraction. The final results are snippets in salient tweets. Our framework differs from (Imran et al., 2013) in which we rank tweets based on their importance based on event extraction. The extraction tries to capture important information such as the time of the event, the number of victims. After extracting, tweets are presented in an event graph, where their importance is computed for ranking. Our framework also shares the classification as well as data in (Imran et al., 2013).

3. Tweet Summarization with Event Extraction and Ranking

This section presents our proposed framework to tackle the challenge of tweets to generate summaries in a disaster scenario. We describe this section in three steps: data preparation, framework, and the summary process.

3.1. Data Preparation

New section

We use 230,535 tweets collected during Joplin tornado in the late afternoon of Sunday, May 22, 2011[2] at Missouri for our experiments. Unique tweets were selected by Twitter Streaming API using the hashtag *#joplin*. The dataset is a part of AIDR project (Imran et al., 2014).

Table 1: The training data and percentage of valuable classes.

Class	**Training examples**	**Percentage (%)**
Informative Information	4,335	—
–Direct Tweets	1,150	—
—Casualty	137	10
—Caution	438	50
—Donation	203	16
—Information source	278	18
—Other	—	6

The training dataset from (Imran et al., 2013) was manually created by using CrowdFlower[3], a crowdsourcing platform that works across multiple crowdsourcing services including Amazon's Mechanical Turk. The authors post a set of tweets into the service and ask crowdsourcing workers to annotate these tweets with predefined instructions. A small number of tweets after annotating is selected by the authors as training data. It includes 6,541 training examples. The inter-annotator agreement for this task is 74.16%. Table 1 summarizes the statistics of training data. The two left columns present classes and their training examples. We can observe that they are organized in three levels: informative, direct, and valuable tweets. The training examples of the first and second classes are quite large, while the number of annotated tweets of the third one is quite small. However, for traditional classification methods such as Max-

[2]http://en.wikipedia.org/wiki/2011_Joplin_tornado
[3]http://www.crowdflower.com

imum Entropy, these training examples are acceptable to train classifiers. We do not consider 6% of other tweets as (Imran et al., 2013) because they are not in the valuable classes.

3.2. Tweet Distilling Framework

New section

A straightforward method to select important tweets from a set of original ones is ranking based on graph. In this graph, vertices are the original tweets and the weight of edges is lexical similarity among tweets. To calculate the similarity Cosine similarity can be considered. However, the noise of tweets challenges this calculation. An interesting idea is to represent a tweet in the form of an event, which is a temple including important snippet information, e.g. *subject*, *action*, *location*, and affected *number* (of damages). The above information supports to answer common questions related to a fact of the disaster, i.e. *what*, *where*, and *how many*? This representation not only helps to avoid the noise of tweets, e.g. spelling errors in informal communication, but also keeps the semantics of tweets in the form of important information. Subsequently, we can again use Cosine to compute the weight of each vertex. Finally, a ranking algorithm can be utilized over the graph to select top m events (having the highest score) corresponding to m original tweets as summaries.

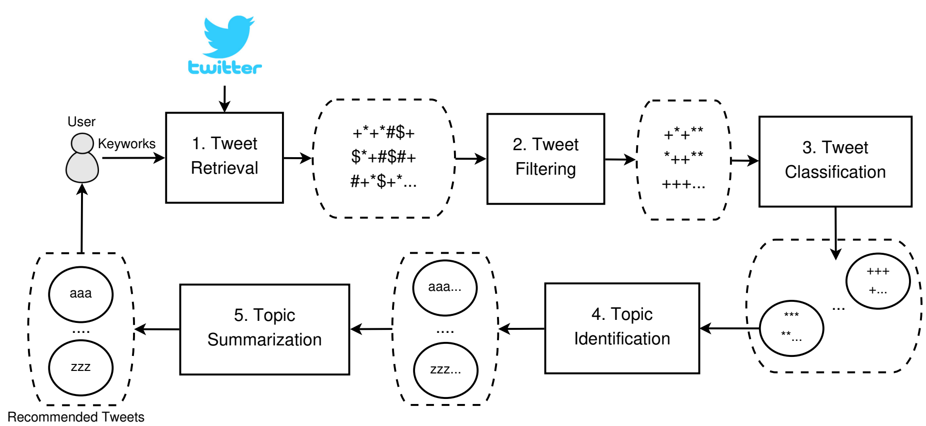


Figure 1: The overview of our proposed framework.

From this idea, we present our proposed framework in Figure 1, which consists of five components. 1) Tweet retrieval retrieves tweets containing a query from the source. 2) Informative filtering selects a small set of the informative tweets because tweets are diverse and noisy with a lot of unrelated ones, e.g. personal information. 3) Tweet classification provides finer-grained information to users, in which it divides informative tweets into predefined classes, i.e. casualties, cautions, and donations for users' ease of navigation. 4) Topic identification separates tweets in the classes into topics denoted in form of clusters for reducing the tweet volume size while ensuring the diversity. Finally, 5) topic summarization distills each topic to produce summary tweets to users. These tweets are called as *informative tweets* or *actionable tweets* thanks to their useful information which helps people making reasonable decisions in a disaster. Note that, given a set of queries, Steps 1 to 5 can be repeated to "monitor" the status of the disaster. The first component is rather simple, therefore, this paper focuses on the four remaining ones.

3.2.1. Tweet Filtering

New section

Our first effort is to remove noise such as irrelevant tweets. This is because even tweets collected by the keywords, there still exist tweets which are not informative. We formulate this task as a binary classification, in which a tweet can be *informative* or *non_informative*. From our investigation, *informative tweets* have two types: *direct* viz. events that users see or hear directly and *indirect*, e.g. a tweet that is forwarded, or *retweeted* in the context of Twitter, or it refers to another source of information. The example of these tweets is shown in Table 2. Obviously, indirect tweets are redundant and not live; therefore, the framework removes them to "compress" the results as an additional preliminary step. More precisely, a tweet is sequentially passed to two classifiers in order to determine whether it is a direct informative one.

Copies of data were annotated by a binary value: 1 and 0 for training binary classifiers. In this step, there are two classifiers, one for identifying informative tweets, and the other for detecting direct tweets. In the first data copy, informa-

Table 2: An example of tweets in valuable classes.

Class	Tweet
Casualty	I live in Joplin, Mo where the F5 tornado hit, 122 dead, On ground for 6 miles. Destroyed hospital, high school, middle school, down town.
Caution	Tornado Warning :: A tornado warning has been issued for Henderson till 10:15pm. Please take appropriate precautions...
Donation	RT Walmart: Weve directed truckloads of water, food and other basic items to the #Joplin area to help the ... http://tmi.me/aBzKU
Direct	@spann: Everybody on the campus of the University of Oklahoma should be in a tornado safe place now. #okwx
Indirect	@TWCBreaking: #Tornado reported in the Kansas City metro! TAKE COVER in Overland Park Leawood Lenexa! Alert!!!

tive tweets were annotated by 1 (informative), otherwise 0 (non-informative) for training the first classifier. In the second data copy, direct tweets were labeled by 1 (direct), otherwise 0 (indirect) for training the second classifier.

3.2.2. *Tweet Classification*

New section

After the first effort to collect a large number of informative direct tweets, this step divides them into *valuable* classes. Though (Vieweg 2012; Khan et al., 2013) indicated that tweets can be divided into 32 valuable classes, we focus on three main classes: casualty/damage, caution/advice, and donation/offer, which support people to make decisions, rather than other classes containing images or videos. In this view, we eliminate tweets in information source and other categories in Table 1.

We use three binary classifiers for detecting casualty, caution and donation. If a tweet is recognized as casualty by the first classifier, it will be passed to the second one to check whether it is a caution tweet. If not, it is passed to the third classifier. In case of a tweet belongs to two classes, its probability predicted by the corresponding classifier is used to decide its class. Three copies of direct tweets were tagged for training the above ones. For example, in the copy of the casualty class, tweets mention casualties or damages were annotated by 1 (casualty); otherwise was 0 (no-casualty). Similar tagging was applied to other data copies.

3.2.3. Topic Identification

The diversity of informative direct tweets in the valuable classes can be expressed in a number of topics the tweets mentioned; therefore, our final effort to preserve this aspect is to assign tweets into topics in the form of clusters. The intuition is that even the number of tweets in each class is not so large, directly extracting a subset of tweets from the original ones ignores important information. Assigning tweets into clusters also helps users to easily navigate the information as well as compress the data. This section describes topic identification in two steps: (i) Latent Dirichlet Allocation (LDA) for generating document probability distribution over topics and (ii) clustering.

Latent Dirichlet Allocation. LDA is one of methods to discover hidden topics in a text collection (Blei et al., 2003; Blei 2012). It is a generative model that allows sets of observations to be explained by unobserved groups, i.e. hidden topics. It explains why some parts of the data are similar in term of the same topic. If the observation bases on words collected from documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. Table 3 shows an example of word distribution over topics after running LDA on our dataset.

Table 3: Some hidden topics with topical words.

Topic 1	**Topic 2**	**Topic...**	**Topic k**
tornado, Joplin, Missouri, 124, ...	Joplin, dead, devastating, ...	...	Oklahoma, kills, 89, death, ...

The hidden topics are denoted by word and document distribution over topics. Therefore, an interesting idea is that the document distribution can be used for assigning documents into clusters (Blei 2012; Khan et al., 2013). However, using LDA faces the selection of a suitable number of topics (k) to reflect the actual diversity of tweets. This can be solved by cluster validation (Levine and Domany 2001; Niu et al., 2007; Brody and Elhadad 2010).

Cluster validation: The idea of cluster validation is to identify the most suitable value of k for topical clusters. The cluster validation finds $k \in I$ (a finite

New section

set of possible number of topics) so that clusters are stable. Suppose that T is a set of tweets, and $T' \subset T$, which is randomly selected (Brody and Elhadad 2010; Niu et al., 2007) in Eq. (1).

$$| T' |= \mu \mid T \mid \tag{1}$$

where μ is proportional parameter satisfying $0 < \mu < 1$. Given two tweets $t_i, t_j \in T$ and $t_i, t_j \in T'$, the clusters are stable if t_i and t_j are assigned into the same cluster when running LDA on T and T'.

Let $C = |T| \times |T|$ be a connectivity matrix to mark if two tweets t_i and t_j include the same topic (e.g. $C_{i,j} = 1$), otherwise $C_{i,j} = 0$; $C' = |T'| \times |T'|$ be the matrix to mark if two tweets t_i and t_j include the same topic (i.e. $C'_{i,j} = 1$), otherwise $C'_{i,j} = 0$. Eq. (2) *proportion of stability* (Levine and Domany 2001), which measures the stability of k clusters.

$$F_k(C', C) = \frac{\sum_{ij} 1(C'_{ij} = C_{ij} = 1, t_i, t_j \in T')}{\sum_{ij} 1(C_{ij} = 1, t_i, t_j \in T')} \tag{2}$$

The best k value maximizes F_k. However, Eq. (2) is bias when k is small (i.e. k = 1), so all tweets belong to a cluster, then the value of F_k is always maximal. To reduce the bias, we employ cluster validation (Brody and Elhadad 2010). The algorithm is shown in Alg. 1, where q is the number of validated times.

The cluster validation first generates a connectivity matrix C by using LDA and R by using a random method. In iterations, it randomly takes a subset of tweets T' from T to construct the connectivity matrix C' while R' is randomly created. After that, stable proportion is calculated by using Eq. (2) over the matrices. The optimal value of k is identified by Eq. (3).

$$k^* = \arg\max_{k \in I}(sta\_pro) \tag{3}$$

The complexity of Algorithm 1 depends on k and q as well as connectivity matrix generating algorithm.

Topical distribution retrieval: After applying LDA with selected k, the framework retrieves tweet distributions over topics, i.e. the topics included in a tweet with a certain probability. Table 4 presents an example of the tweet probabilistic

New section

Algorithm 1: The cluster validation method.

Input: a set of number topics and original tweets;
Output: the number of topics;
for $k \in I$ **do**
  Generate initial connectivity matrix C using LDA;
  Generate initially random connectivity matrix R;
  for i = 1 to q **do**
    Selecting T' by Eq. (1);
    Generate connectivity matrix C' using LDA;
    Generate random connectivity matrix R';
    $sta\_pro \leftarrow F_k(C', C) - F_k(R', R)$ by Eq. (2);
  Find k by Eq. (3);

Table 4: The example of tweet probability distribution over topics. Rows are tweets and columns present topics.

0.270	0.050	0.210	0.015	0.230	0.035	0.100	0.090
0.080	0.075	0.045	0.170	0.190	0.240	0.080	0.120
0.130	0.120	0.180	0.105	0.260	0.096	0.033	0.076
...	...	...	...	...	...	...	...
0.068	0.114	0.215	0.037	0.138	0.055	0.081	0.292

distribution of T tweets and k topics. In Table 4, row i^{th} is the tweet probability distribution over topics generated from LDA, column j^{th} is the tweet distribution over topic j^{th}, and value at an element (i, j) represents the tweet distribution of tweet i^{th} over topic j^{th}.

Clustering. Traditional clustering methods such as k-means can be directly applied to assign original tweets into clusters; however, they base on term frequency, which ignores topical aspect mentioned in tweets. Intuitively, tweets in the same cluster not only share common words but mention common topics. We, therefore, propose to utilize hidden topic models to capture the topical aspect. The intuition is that tweets mentioning common topics should be assigned into the same cluster. In this sense, our method can be regarded as topic-driven summarization (Ma et al., 2012; Khan et al., 2013; Yajuan et al., 2012).

In running LDA, each tweet was considered as a document and presented by a probability distribution over topics. It is possible to assign tweet i^{th} in row i^{th} in a topic j^{th} by selecting maximal value in each row. For example, in Table 4, tweet 2^{nd} denoted by row 2^{nd} can be assigned into topical cluster 6^{th} with maximal value = 0.240. However, this method has no explicit mechanism to measure the *similarity* of a new tweet giving a current topical cluster. We, therefore, utilize distance calculation between tweet probability distribution over topics and clusters to assign a tweet into a real cluster. In this setting, the number of clusters equals to the number of topics.

We adopted Jensen-Shannon divergence[4] to calculate the distance between two probability distributions for clustering (Khan et al., 2013). Let $C = k \times k$ be a matrix representing clusters and is initialized by a unit matrix; and $DP =\mid T \mid \times k$ (in Table 4) is the topical distribution of tweets generated by LDA. Let C_i be the i^{th} row of the matrices C and DP_j is a row in DP. Eqs. 4, 5, 6, and 7 show the clustering:

$$z = \underset{i \in [1,k]}{\operatorname{argmin}}(D_{JS}(C_i, DP_j)) \tag{4}$$

The value of z represents the shortest distance from a tweet j^{th} to a cluster c_i. The $D_{JS}()$ is Jensen-Shanon divergence computed in Eq. (5).

$$D_{JS}(A_i, DP_j) = \frac{1}{2}(D_{KL}(C_i||M) + D_{KL}(DP_j||M)) \tag{5}$$

$$M = \frac{1}{2}(C_i + DP_j) \tag{6}$$

where M is the mean point of two probability distributions. The $D_{KL}()$ in Eq. (5) is Kullback-Leibler Divergence[5], which measures the information loss when

[4] https://en.wikipedia.org/wiki/JensenShannon_divergence

[5] https://en.wikipedia.org/wiki/KullbackLeibler_divergence

using Q to approximate P. For discrete probability distributions Q and P, the equation is shown in Eq. 7:

$$D_{KL}(P||Q) = \sum_i \ln \left(\frac{P(i)}{Q(i)} \right) P(i) \tag{7}$$

With the calculation in Eqs. (4), (5), (6), and (7), we present an algorithm for clustering in Alg. 2.

Algorithm 2: Assigning tweets into clusters.

Input: topical number and tweet distribution matrix;
Output: clusters including similar tweets;
for DP_j *in* DP **do**
  for C_i *in* C **do**
    $M \leftarrow calculatingM(C_i,\ DP_j)$ by Eq. (6);
  for C_i *in* C **do**
    $distance = JSDiv(C_i,\ DP_j,\ M_i)$ by Eq. (5);
    $L \leftarrow distance$;
  $pivot = findMinValue(L)$;
  $assigning(DP_j,\ pivot)$;
  $C[pivot] = M[pivot]$;

Function $calculatingM()$ first calculates a mean point between DP_j and C_i by using Eq. (6), then computes the distance from tweet j^{th} to clusters by using Eq. (4). Finally, the algorithm finds a cluster having the closest distance and updates the distribution of cluster i^{th} corresponding to row i^{th} in matrix C. The complexity of the clustering depends on the number of tweet T and the number of topical cluster k.

3.2.4. Topic Summarization

The last step is to distill every topic generated in the clustering to find out a small set of actionable tweets in each cluster as extractive summarization. The idea of this step is to rank the tweets in a topical cluster, remove near duplicate ones, and get m top ranked ones as the summary tweets. Normal ranking algorithms only use keywords or even though topical statistics can not completely utilize the semantics of tweets and do not effectively exploit the similarity among tweets within a cluster. We, therefore, propose to represent a tweet in the form of an event which is the sketch of a tweet to avoid the noise of lexical representation while preserving the semantics of tweets. Concretely, we carried out the following steps: 1) applying event extraction to represent tweets; 2) constructing an event graph to preserve the convergence of similar tweets; 3) ranking the graph to achieve high ranked tweets, 4) removing near duplicate tweets to return m top ranked (the highest score) to users.

Event extraction. We define an event as a set of attributes (for answering common questions) in a tweet, namely, *subject*, *action*, *location*, and *number*:

$$event = \{subject, action, location, number\} \tag{8}$$

where *subject* answers the question WHAT, e.g. a tornado or a road, which is a cause or result; *action* represents the action/effect of the subject; *location* answers WHERE the event occurred, e.g. Oklahoma; and *number* answers the question HOW MANY, e.g. the number of victims.

To extract the above attributes, we employed a NER tool (Ritter et al., 2011) in which tweets are annotated by predefined tags; then, they are parsed to extract values of tags corresponding to the attributes. More precisely, *subject* is extracted by words/phrases labeled by *"NN"*; *action* is captured by words/phrases labeled by *"B-EVENT"*; *localtion* is extracted by words/phrases labeled by *"B-geo-loc"*; and *number* is captured by words/phrases labeled by *"IN"* and *"CD"*. We accept an event which does not have full attributes. An example of an event from an original tweet is shown as below:

Original tweet: *"Tornado kills 89 in Missouri yesterday".*

Event: {*Tornado, kills, Missouri, 89*}.

In this example, *Tornado* is the subject, *kills* is the event phrase, *Missouri* is the location, and *89* is the number of victims.

Event graph construction. Each event of a tweet is a node in an event graph. To construct the edge and its weight, we consider two measurements: Cosine and Simpson. Let A and B are two events, they are first converted into vector space based on bag-of-words model, subsequently, Cosine similarity of the two events was calculated by Eq. (9).

$$cosine(A, B) = \frac{\sum_i A_i \times B_i}{\sqrt{\sum_i (A_i)^2} \times \sqrt{\sum_i (B_i)^2}} \tag{9}$$

where A and B are the same size vectors. Simpson is computed by Eq. (10):

$$simp(A, B) = 1 - \frac{|S(A) \cap S(B)|}{min(|S(A)|, |S(B)|)} \tag{10}$$

where $S(A)$ and $S(A)$ are the sets of words of A and B. The value of these equations ranges from 0 (totally different) to 1 (identical). Since an event consists of only a few words, after an investigation, we observe that Cosine is more precise than Simpson. Table 5 shows an evidence, where the value of *1.0* of Simpson indicates that two tweets are identical despite the difference of the word *"yesterday"* between the two sentences, whereas the value *0.912* of Cosine indicates the difference. The threshold to decide whether there is an edge between two events is 0 (there is an edge if $cosine(.) > 0$). Isolated nodes (node has no edge) are removed since they are treated as noise in a cluster. The $cosine(.)$ value is also used as the weight of the edge. Concretely, we define the graph of a cluster as an undirect graph of $G = \langle V, E, W \rangle$ where:

Table 5: The illustration of two equations.

Tweet	Simpson	Cosine
Tornado kills 89 in Missouri. Tornado kills 89 in Missouri yesterday.	1.0	0.912

- $V = \{v_1, v_2, ..., v_n\}$ is a set of vertices, where v_i^{th} corresponds to an *event* i^{th} and n is the number of events having $cosine(.) > 0$ to at least one event in the cluster.

- $E = \{e_1, e_2, ..., e_m\}$ is a set of edges, where e_j^{th} connects two vertices v_k^{th} and v_h^{th} in V.
- W is the weight matrix of edges with $W_{ij} = cosine(v_i, v_j)$.

Ranking. We employ PageRank (Brin and Page 1998), a ranking algorithm that can exploit the relation among objects in the form of a directed graph, to our graph due to its efficiency in ranking Web pages. The detail of this algorithm is described as follows:

New section

PageRank definition: Let $E(u)$ be a vector over the Web pages that corresponds to a source of rank. Then, the PageRank of a set of Web pages is an assignment, R', to the Web pages that satisfies

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u) \tag{11}$$

such that c is maximized and $||R'||_1$=1 ($||R'||_1$ denotes the L_1 norm of R', i.e. $L_1 = \sum_{u \in Webpages} R'(u)$), where B_u is the set of incoming edges of u, and N_v is the total number of outgoing edges to v.

Let A be a square matrix of $n \times n$, where n is the total number of Web pages, such that $A_{u,v} = 1/N_u$ if there is an edge from u to v, otherwise, $A_{u,v} = 0$. Then, from the Eq. (11), we have $R' = c(AR' + E)$. Since $||R'||_1 = 1$, we can rewrite as $R' = c(A + E \times 1)R'$, where 1 is the vector of all ones. So, R' is an eigenvector of $(A + E \times 1)$. The computation of R' is straightforward if the scale issue is ignored. Let S be almost any vector over Web pages (e.g. it is possible to set $R' = E$), then, Alg. 3 can be used to approximately estimate R' with a small threshold ϵ to control the error level.

In the context of web-pages, the relation (i.e., the link) is represented by a directed edge. However, the event graph in our study is undirected graph; therefore, to apply the PageRank, the relation was simply treated to be undirected,[6] i.e. if $W_{ij} > 0$, then there is an edge from v_i to v_j, and another from v_j to v_i

[6] http://en.wikipedia.org/wiki/PageRank#PageRank_of_an_undirected_graph

Algorithm 3: PageRank algorithm.

$R_0 \leftarrow$ S;
repeat
  $R_{i+1} \leftarrow AR_i$;
  d $\leftarrow ||R_{i+1}||_1$ - $||R_i||_1$;
  $R_{i+1} \leftarrow R_{i+1}$ + dE;
  $\delta \leftarrow ||R_{i+1}$ - $R_i||_1$;
until $(\delta > \epsilon)$;

with the same weight of W_{ij}. This means that all edges of a vertex v_i having $W_{ij} > 0$ are considered as both inlinks and outlinks.

Post processing. Tweets in the same cluster have similar content, to ensure the diversity in the final results, one more step was added to keep unique ones. Though it is possible to apply this step before ranking, it was put after to avoid a so sparse event graph, that may negatively affect the ranking.

Since an event is just the sketch of a tweet, it is not suitable for evaluating the duplication based on the event as shown in Table 6 where two tweets mention two different facts but they have the same event form. Therefore, original tweets are used with Simpson calculation for comparing tweets as shown in Table 5.

Table 6: An example of events and tweets.

Tweet	Event
Tornado hit Missouri yesterday	{Tornado, hit, Missouri}
Many people was died at Missouri by a tornado	{Tornado, died, Missouri}

Two tweets are deemed to be near-duplicate if the $simp(.)$ in Eq. (10) is greater than a certain threshold.

4. Statistical Analysis

This section first presents the setup for our experiments. It next introduces New section baselines used to compared to our framework and shows the evaluation method.

4.1. Parameter Setup

An open source of LDA tool[7] was used (Phan et al., 2008); $\alpha = \beta = 0.01$ with 1.000 iterations; and $k \in [2, 50]$ is identified in §5.2. $\mu = 0.9$ was used in Eq. (1). The threshold of Eq. (10) for keeping tweets was 0.25, i.e. $simp(t_1, t_2) > 0.25$ by running the experiments over several times. A variation of PageRank applying for undirected graphs was utilized.[8] NER tool for tweets[9] (Ritter et al., 2011) was also used to extract event's elements. We extract $m = 10$ tweets as the summarization for each cluster.

For the classification, we employed Maximum Entropy (ME)[10] to build binary classifiers with n-gram ($n = 1, 2$) features(Nigam et al., 1999; Ratnaparkhi 1996; Rosenfeld 1996). Also, our data is sparse after pre-processing, and ME has shown its efficiency at dealing with sparse data (Phan et al., 2008). Hashtags, emoticons, or retweets information were not used because they are usually utilized in emotional analysis rather than classification. The classification was evaluated by 10-folds cross-validation.

4.2. Baselines

We compared our framework to three baselines, in which one bases on retweet counting and the two remaining methods are derivations from ours.

Retweet. Retweet model was used as a baseline (Busch et al., 2012) because it measures the importance of a tweet based on its retweet counting. Intuitively, if a message receives many retweets, it can be considered to be informative. The model first receives tweets which belong to clusters after clustering, then ranks tweets based on the number of retweets by a ranking algorithm. For this

[7] http://jgibblda.sourceforge.net/
[8] https://github.com/jia1546/PageRank/tree/master/src/pagerank
[9] https://github.com/aritter/twitter_nlp
[10] http://www.cs.princeton.edu/maxent

mechanism, tweets having the highest retweets are in the top of ranked list. Finally, the model takes top 10 tweets from the ranked list to provide for users.

No Event Extraction *(No-extraction).* A model that dose not utilize event extraction was used as another baseline to prove the role of event extraction. This model also receives tweets from clusters as input, then ranks these tweets based on a variation of PageRank algorithm. This method uses original tweets to calculate Cosine similarity instead of events. Finally, the model selects top 10 tweets in each cluster to recommend for users after near-duplicate removal.

No Ranking *(No-ranking).* is another our effort to investigate the effectiveness of ranking in Topic summarization. To do that, we developed the third baseline in which at the topic summarization step, 10 tweets were randomly selected to return to users, instead of using event extraction and ranking.

4.3. Evaluation Method

User's rating. To evaluate the performance of the framework, three annotated groups including nine annotators with good English skill were asked to rate top m (m=10) tweets. This is because there is no gold-standard references in the dataset. The annotators rate a value from 1 to 5, in which 5: very good; 4: good; 3: acceptable; 2: poor; and 1: very poor on extracted tweets. The score measures how much informative information of a tweet provide for readers. Each tweet is rated by three reviewers; the score of a tweet is the average of rating scores. For example, tweet: *"I live in Joplin, Mo where the F5 tornado hit, 122 dead, On ground for 6 miles. Destroyed hospital, high school, middle school, down town."* was rated by 5 because it mentions a lot of information about the casualty: the number of death (122), the level of tornado (F5 and 6 miles), and the damage (hospital, high school, and downtown), while tweet: *"Last time a tornado touched down in Springfield Mass was 1972. That was the last time."* only mentions the time when the last tornado appears (1972), then it was rated by 1.

New section

After finishing the rating step, results were validated by using cross-validation method. Each annotated tweet is given to other annotators in the same group to check the agreement. If annotators agree with the prior value, this tweet is labeled YES; otherwise, it is assigned by NO label. The average of agreement in each cluster is computed by Eq. (12).

$$Inter - rating\ agreement = \frac{\#YES}{\#tweets} * 100\ (\%) \tag{12}$$

where $\#YES$ is the number of tweets rated by YES and $\#tweets$ are total annotated tweets. The average of inter-rating agreement after cross-checking is shown in Table 7.

Table 7: The detail of inter-rating agreement over three annotated groups.

Annotators	**Casualty**	**Caution**	**Donation**	**Average (%)**
Group 1	82.4	79.8	91.2	84.46
Group 2	78.7	80.3	87.6	82.53
Group 3	84.5	86.1	90.6	87.06
	Inter-rating agreement			84.68

Evaluation metric. To evaluate the classification, precision (P), recall (R) and F-score (F-1) were used. To evaluate the performance of the summarization, ROUGE-scores (Lin and Hovy, 2003) can be used. It matches extracted tweets to gold-standard references to compute $n-$grams overlapping. However, as mentioned, the references are unavailable in this dataset. Therefore, we define *completeness* to measure how well the summary covers the informative content in the extracted tweets. It is computed by the division of total rated scores over maximal score in each cluster.

$$completeness = \sum \frac{rating\ score}{50} * 100\ (\%) \tag{13}$$

where *rating score* is the users' score, and 50 is maximum total score viz. 10 tweets each has a maximum score of 5.

5. Results and Discussion

This section first shows the results of classification in §5.1, following by the sensitivity analysis of selecting topic number k in §5.2. The summary results are presented in §5.3. We finish with an error analysis in §5.4.

5.1. Tweet filtering and classification

We first report the performance of the filtering and classification in Table 8. The performance of classifying informative and direct tweets is quite poor even though the training examples are large. Because in the first level, tweets are very noise; hence, identifying the informative ones is very challenge. This is also the same in the second level. In addition, using n-gram features also limits the classification. It is possible to integrate additional refined features for characterizing informative and direct tweets. The remaining levels obtained acceptable results for the later steps. The classification performance is similar to (Imran et al., 2013) because we share the training data but use different training algorithm.

Table 8: The performance of classification

Class	**Precision**	**Recall**	**F1-score**
Informative Information	0.75	0.87	0.80
–Direct Tweets	0.71	0.83	0.77
—Casualty	0.89	0.89	0.89
—Caution	0.88	0.91	0.89
—Donation	0.87	0.88	0.88

5.2. Cluster Validation

As mentioned, we use cluster validation to find out an appropriate k for each valuable class. Precisely, we tuned k in $2 \leq k \leq 50$ because if $k > 50$ clusters may be over-fitting while it is too general if $k < 2$ (only one cluster). Figure 2 presents the sensitivity analysis of selecting k.

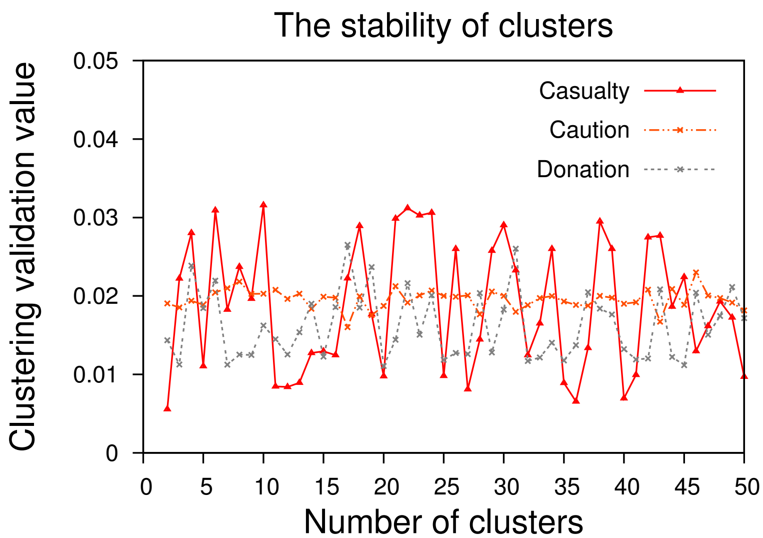


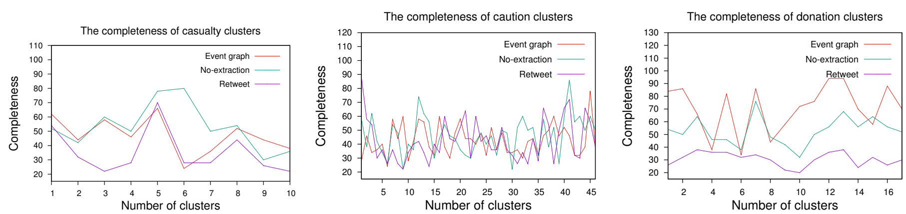
Figure 2: Cluster validation on various k numbers.

The values in Figure 2 show that changing k affects the validation. While the general trend of caution and donation is quite stable with small margin between maximum and minimum values, it is hard to observe the tendency in casualty. This may originate from the noise of tweets. Also, the small number of tweets in casualty and donation can make a challenge to measure their stability. Based on Figure 2, we set $k = 10$ for casualty, $k = 46$ for caution, and $k = 17$ for donation class. After clustering, informative tweets belonging to clusters are put into the summarization.

5.3. Tweet Summarization

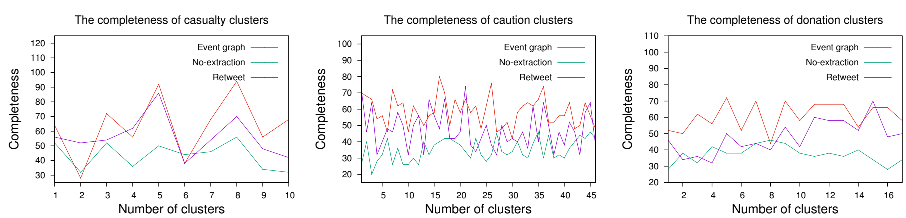
This section presents the comparison in two scenarios. We first report the summary performance of our framework against the two baselines: no-extraction and retweet to validate the efficiency of our model. It next shows the comparison of using ranking (our framework and no-extraction method) to no-ranking to reveal role of ranking in the summary process.

In the first scenario, we report the completeness of our framework and the two baselines: no-extraction and retweet in Figures 3, 4, and 5. The completeness from these figures indicate that our framework clearly outperforms the baselines in Figures 3c, 4b, 4c, 5a, 5b and comparably performs in the remaining figures. This shows that our proposal is efficient for extracting informative tweets by using event extraction. For example, in Figure 4c, the completeness of our framework surpasses no-extraction and retweet in almost clusters. This is because the framework exploits event extraction, which can enrich the semantics and reduce the noise among tweets. The trend of our method and the no-ex-



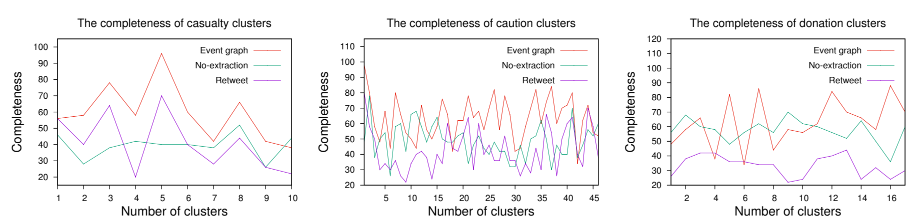
(a) Casualty class (b) Caution class (c) Donation class

Figure 3: The results of three methods with the 1^{st} annotator group.



(a) Casualty class (b) Caution class (c) Donation class

Figure 4: The comparison of three methods with the 2^{nd} annotator group.

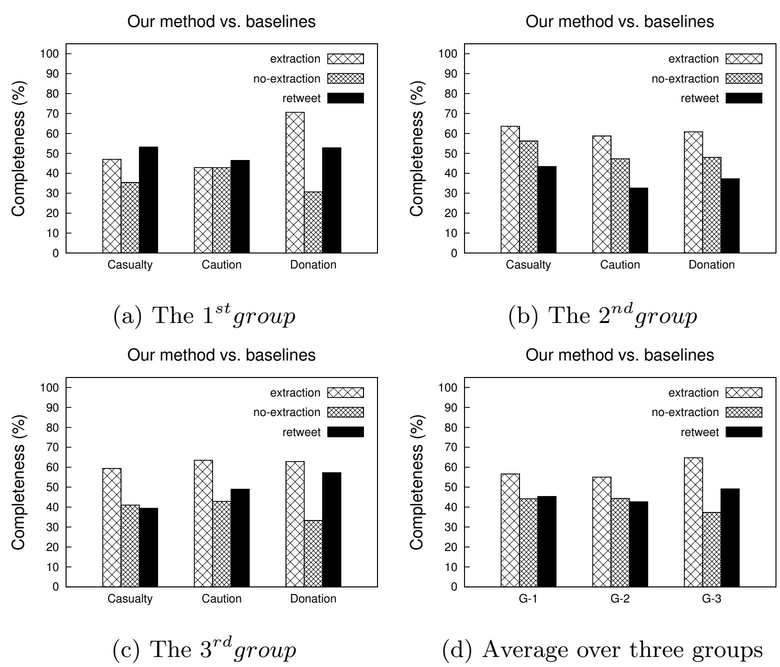


(a) Casualty class (b) Caution class (c) Donation class

Figure 5: The comparison of three methods with the 3^{rd} annotator group

traction indicates that our method significantly outperforms the no-extraction (Figures 6b and 6c) even they share the ranking algorithm. This is because the no-extraction does not utilize the extraction, which challenges the Cosine calculation due to the noise of tweets. In some cases, our method comparably performs the baselines because the evaluation method is objective based on the rating of users; therefore, it is hard to obtain the same conclusion of a tweet among annotators. This is possibly solved by generating a set of gold-standard references to evaluate the summarization. Interestingly, retweet model outputs better results than the non-extraction in some cases such in Figure 6c. This is because user interests on tweets are a valuable aspect to judge their important. It suggests that user aspect can be exploited to improve the quality of ranking (Yajuan et al., 2012).

To easily observe the comparison, we provide the average of completeness over each group and over the three group. Figures 6a, 6b, and 6c show the average of completeness over each group whereas Figure 6d reports the average over the three groups. The results again support the trend in Figures 3, 4, and



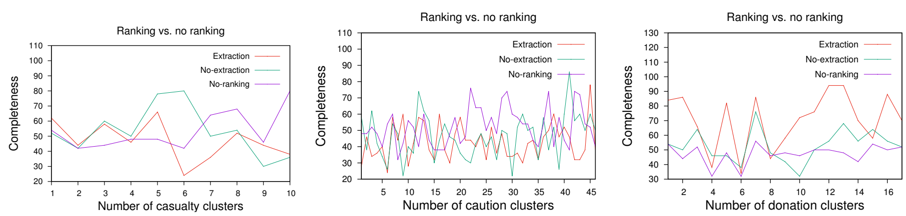
(a) The $1^{st} group$ (b) The $2^{nd} group$ (c) The $3^{rd} group$ (d) Average over three groups

Figure 6: The average completeness of three methods rated by three groups.

5, in which our framework clearly outperforms the baselines in almost cases. In Figure 6d, our method significantly outperforms the two baselines of 12% and

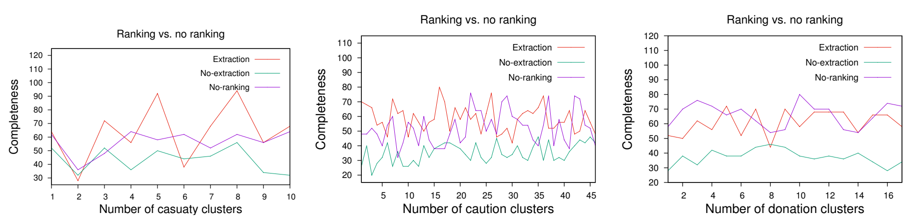
17%. The retweet is competitive with ranking without event extraction although it is a simple method. This is because, in some cases, important tweets receive a lot of attention from readers leading a high number of retweet.

In the second scenario, we present the comparison of ranking vs. no-ranking in Figures 7, 8, and 9. The completeness indicate that the ranking with event



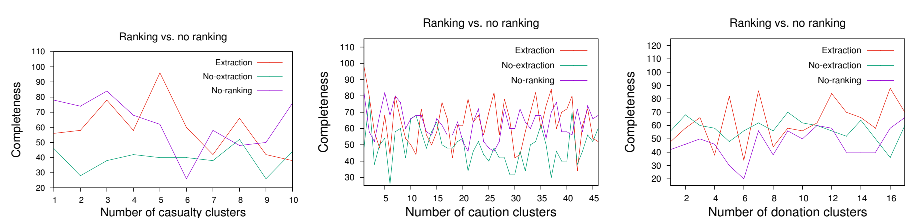
(a) Casualty class (b) Caution class (c) Donation class

Figure 7: Ranking vs. no ranking of the 1^{st} user group.



(a) Casualty class (b) Caution class (c) Donation class

Figure 8: Ranking vs. no ranking of the 2^{nd} user group.



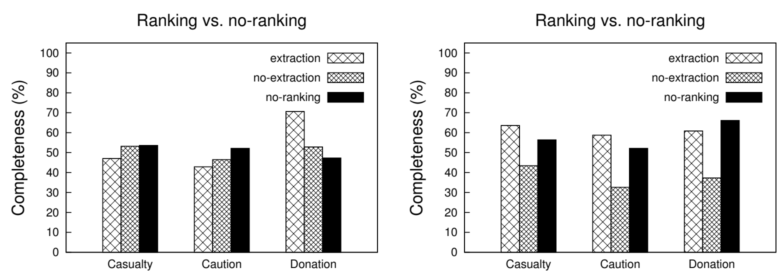
(a) Casualty class (b) Caution class (c) Donation class

Figure 9: Ranking vs. no ranking of the 3^{rd} user group.

extraction clearly outperforms the baselines in Figures 7c, 8b, 9b, and 9c and is competitive in the remaining ones. It outputs better results in Figure 8c than the random method. This is because some informative tweets can be randomly picked up, as a result, no ranking may achieve high completeness, in some cases.

Also, tweets in the same cluster tend to be similar in term of content, therefore, the random selection provides a reasonable mechanism for selecting tweets.

To facilitate the observation, we provide the average of completeness over each group as well as over the three group. From Figures 10a, 10b, 10c we can observe that the extraction with ranking (our method) significantly outperforms the ranking without the extraction (no-extraction). This supports our idea stated in §3.2, in which event extraction benefits the ranking. The no-ranking



(a) The average of $1^{st} group$

(b) The average of $2^{nd} group$

(c) The average of $3^{rd} group$

(d) The average over three groups

Figure 10: The average completeness of ranking vs. no ranking.

(random method) obtains competitive performance, in which it is even better than our framework such in Figure 10a in casualty and caution. It is encouraged in disaster scenarios because a simple method can output high-quality tweets over a sophisticated clustering algorithm. The average over three groups is also shown in Figure 10d. This indicates that our method is 12% better than no-ranking in *group 3* and is competitive in the two remaining ones.

5.4. Error Analysis

We take a look into the data and show discuss in some outlier clusters. Those which have a high completeness contain highly relevant tweets appearing in all the methods. Almost top 10 tweets of our method in cluster 1^{st} in Figure 5b, 5^{th} in Figure 5a; and 41^{th} of no-extraction; and 3^{rd} in Figure 9a or 10^{th} in Figure 8c are strongly related to the cautions, advices, or casualties of the tornado. For example, tweet: *"Tornado on the ground in the Metro–touching down in South JoCo. Get to the basement, folks!"* was rated by 5 because it mentions the fact of tornado and provides an advice. By contrast, low completeness clusters include irrelevant tweets, including images and videos or tweets about President Obama visiting Joplin. For example: *"RT ALB24: "kevin thornton: Pics of hail near Dallas. Warning: it's huge. http://yfrog.com/gyq3jxbj http://bit.ly/lhEqIr #TXWX" spann"* with 33 retweets; or *"RT springfieldNL: Obama says no doubt in my mind, Joplin will rebuild. We will be with you every step of the way"*. Therefore, they receive a lot of attention and clusters contain these tweets, e.g. 1^{st} in 3b achieve high completeness (e.g. 0.9).

The Retweet method in donation contains the speech of Eric Cantor in opposing disaster relief and saving money. For instance, tweet: *"Cantor Says Congress Won't Pay For Missouri Disaster Relief Unless Spending Is Cut Elsewhere: The deadly tornado in Jop http://bit.ly/kcNCI1"* receives a lot of retweet leading to reduce the performance of tweet retrieval (cluster 10^{th} in donation in Fig. 3c and 9^{th} in Fig. 5c). In addition, many tweets mention memorial services for victims. For example, tweet: *"Don't forget the Memorial planned in #Joplin. Let us continue to help those in need, especially those recovering from ..."*. These kind of tweets also receive many retweets leading low performance of *Retweet* compared to our approach.

The performance of no-extraction is lower than our method due to the calculation, e.g. cluster 9^{th} in Figure 5a. For example, tweet: *"Please Pray for the people of Joplin a tornado is on the ground near there!!!!"* does not mention casualty, but this tweet contains words e.g., people, Joplin, tornado, ground which appear in many tweets. Therefore, the performance of the no-extraction is reduced.

Cluster 5^{th} of caution in Figure 3b is the lowest completeness of our method. The checking process shows that many events are irrelevant indicating clustering and event extraction are inefficient in this cluster. For example, the event having the highest value contains only two entities *pic* and *tonardo*; but obviously, this event is not enough evidence to calculate the similarity with others because it is too short. It is possibly solved by a sophisticated event similarity calculation.

Table 9 presents an example of extracted tweets from our framework. It is clear to observe that they mention important information of casualty, caution, and donation. For example, in the casualty, the first and second tweet provide valuable information of the damage such as 122 dead or destroy the hospital in the first one, and 116 people have died in the second one. In the caution, the second and third tweet can be considered as a warning and require people taking the shelter immediately. In the donation, people can find the needs such as blood in the third tweet. In addition, all the tweets in Table 9 include snippet information we need to create events such as *subject*, *number*. As a result, they receive high scores compared to the others. However, one of important aspect is the truth of messages from social users. If possible, we can integrate a module to judge whether a tweet is truthful to provide better evidence for users in making their decisions.

Table 9: Extracted tweets generated from our framework.

Class	Tweet
Casualty	I live in Joplin, Mo where the F5 tornado hit, 122 dead, On ground for 6 miles. Destroyed hospital, high school, middle school, down town.
	J#oplin wow that was a huge tornado you only have to look at the time it took to rip apart the town, 116 people have died already.
	We got hit by a tornado. There was a dead and a lot of damage. We are ok.
Caution	Tornado warning! A tornado south of here coming towards us. In the basement.
	Tornado Warning has been issued for Pushmataha County. Take tornado precautions now!
	Tornado Warning for Craighead County for the tornado thats on the ground in Jackson County!! SEEK SHELTER NOW!!! #ARWX.
Donation	Metro & others for tornado help. 722 service hrs moving residents, volunteers. Good work operators! http://ow.ly/5emXH
	Im looking for donations for fundraising help w/ tornado relief. If you can help, email me cmthornton at gmail.com. Anything big or small.
	RT RedCross: Were prepared to help w/blood needs in #Joplin area now. Call 1-800-RED CROSS to help before the next disaster. http://b...

6. Conclusion

This paper presents a distilling framework for retrieving informative tweets during a disaster for suitable reactions. Our framework utilizes state-of-the-art machine learning techniques, event extraction, and graphical model to deal with the diversity, large volume and noise of tweets. The insight behind our framework is to exploit event extraction to enrich the semantics and reduce the noise of tweets in computing their scores. The event extraction allows to present tweets into event graphs, where a ranking algorithm operates to extract salient tweets as the summarization. Experimental results also indicate that by using event extraction, our framework significantly outperforms 3.24%, 13.1% and 16.86% of completeness over three baselines: no-ranking, no-extraction and retweet, correspondingly. Information from our framework suggests that it can be combined with other sources, e.g. TV, online news, or emergency services in dealing with real disasters.

For future directions, firstly, the ranking should incorporate other features such as user aspect (Yajuan et al., 2012) in order to improve its performance. In the post processing step, a sophisticated method in removing near duplicate tweets, e.g. recognizing textual entailment should be integrated. Finally, abstractive and social context summarization for disasters should be considered (Yang et al., 2011; Wei and Gao, 2014; Nguyen and Nguyen, 2016; Nguyen and Nguyen, 2017).

Acknowledgments

This work was supported by JSPS KAKENHI Grant number JP15K16048, JSPS KAKENHI Grant Number JP15K12094, and JST CREST Grant Number JPMJCR1513, Japan. We would like to thank Muhammad Imran at Qatar Computing Research Institute for sharing dataset via the AIDR project; Chien-Xuan Tran for building the Web interface. We also thank the comments of anonymous reviewers for improving our paper.

References

Anderson, C. (2012). Japan Earthquake Social Media Coverage: Disaster By The Numbers. In *Huffington Post*, *9*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*(Jan), 993–1022.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engines. *Computer Networks and ISDN System*, *30*(1), 107–117.

Brody, S., & Elhadad, N. (2010, June). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, (pp. 804–812). Association for Computational Linguistics.

Busch, M., Gade, K., Larson, B., Lok, P., Luckenbill, S., & Lin, J. (2012, April). Earlybird: Real-time search at twitter. In *Proceedings of the 28th International Conference on Data Engineering (ICDE)*, (pp. 1360–1369). IEEE.

Chakrabarti, D., & Punera, K. (2011). Event Summarization Using Tweets. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, (pp. 66–73). AAAI.

Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013, May). Extracting information nuggets from disaster-Related messages in social media. In *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, (pp. 791–800).

Imran, M., Castillo, C., Lucas, J., Meier, P., & Vieweg, S. (2014, April). AIDR: Artificial intelligence for disaster response. In *Proceedings of the 23rd International Conference on World Wide Web (WWW)*, (pp. 159–162). ACM

Jrvelin, K., & Keklinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 422–446.

Khan, M. A. H., Bollegala, D., Liu, G., & Sezaki, K. (2013, September). Multi-tweet summarization of real-time events. In *International Conference on Social Computing (SocialCom)*, (pp. 128–133). IEEE.

Lin, C. Y., & Hovy, E. (2003). Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* (Volume 1, pp. 71–78). Association for Computational Linguistics.

Levine, E., & Domany, E. (2001). Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, *12*(11), 2573–2593.

Ma, Z., Sun, A., Yuan, Q., & Cong, G. (2012, October). Topic-driven reader comments summarization. In *Proceedings of the 21st ACM international Conference on Information and Knowledge Management (CIKM)*, (pp. 265–274). ACM.

Niu, Z. Y., Ji, D. H., & Tan, C. L. (2007, June). I2r: Three systems for word sense discrimination, chinese word sense disambiguation, and english word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, (pp. 177–182). Association for Computational Linguistics.

Nigam, K., Lafferty, J., & McCallum, A. (1999, August). Using maximum entropy for text classification. In *IJCAI-99 workshop on Machine Learning for Information Filtering* , (pp. 61–67).

Nguyen, M. T., Kitamoto, A., & Nguyen, T. T. (2015, May). Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, (pp. 64–74). Springer International Publishing.

Nguyen, M. T., & Nguyen, M. L. (2016). SoRTESum: A Social Context Framework for Single-Document Summarization. In *European Conference on Information Retrieval (ECIR)*, (pp. 3–14). Springer International Publishing.

Nguyen, M. T., & Nguyen, M. L. (2016). Intra-relation or Inter-relation?: Exploiting Social Information for Web Document Summarization. *Expert Systems with Applications*, *76* (Jan), 71–84.

O'Connor, B., Krieger, M., & Ahn, D. (2010, May). TweetMotif: Exploratory Search and Topic Summarization for Twitter. In *the Fourth International AAAI Conference on Weblogs and Social Media (ICWSM)*, (pp. 384–385). AAAI.

Phan, X. H., Nguyen, L. M., & Horiguchi, S. (2008, April). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th international conference on World Wide Web (WWW)*, (pp. 91–100). ACM.

Ratnaparkhi, A. (1996, May). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Vol. 1, pp. 1524–1534). Association for Computational Linguistics.

Ritter, A., Clark, S., & Etzioni, O. (2011, July). Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1524–1534). Association for Computational Linguistics.

Ritter, A., Etzioni, O., & Clark, S. (2012, August). Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, (pp. 1104–1114). ACM.

Rosenfeld, R. (1996). A maximum entropy approach to adaptive statistical language modeling. *Computer Speech & Language*, *10*(3), 187–228.

Rudra, K., Ghosh, S., Ganguly, N., Goyal, P., & Ghosh, S. (2015, October). 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)*, (pp. 583–592). ACM.

Rudra, K., Banerjee, S., Ganguly, N., Goyal, P., Imran, M., & Mitra, P. (2016, July). Summarizing situational tweets in crisis scenario. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media (HT)*, (pp. 137–147). ACM.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, (pp. 851–860). ACM.

Verma, S., Vieweg, S., Corvey, W. J., Palen, L., Martin, J. H., Palmer, M., Schram, A. & Anderson, K. M. (2011, July). Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, (pp. 385–392). AAAI.

Vieweg, S., Hughes, A. L., Starbird, K., & Palen, L. (2010, April). Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI)*, (pp. 1079–1088). ACM.

Vieweg, S. (2012). Situational awareness in mass emergency: A behavioral and linguistic analysis of microblogged communications. *Doctoral dissertation.* University of Colorado at Boulder.

Xu, W., Grishman, R., Meyers, A., & Ritter, A. (2013). A preliminary study of tweet summarization using information extraction. *Proceedings of the Work-*

shop on Language in Social Media (LASM), (pp. 20–29). Association for Computational Linguistics.

Wang, Z., Shou, L., Chen, K., Chen, G., & Mehrotra, S. (2015). Text summarization using a trainable summarizer and latent semantic analysis. *IEEE Transactions on Knowledge and Data Engineering*, *27*(5), 1301–1315.

Wei, Z., & Gao, W. (2014). Utilizing Microblogs for Automatic News Highlights Extraction. In *COLING*, (pp. 872–883). Association for Computational Linguistics.

Wu, K., Li, L., Li, J., & Li, T. (2013). Ontology-enriched multi-document summarization in disaster management using submodular function. *Information Sciences*, *224*, 118–129.

Yajuan, D., Zhimin, C., Furu, W., Ming, Z., & Shum, H. Y. (2012). Twitter topic summarization by ranking tweets using social influence and content quality. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING*, (pp. 763–780). Association for Computational Linguistics.

Yang, Z., Cai, K., Tang, J., Zhang, L., Su, Z., & Li, J. (2011). Social Context Summarization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 255–264). ACM.

Yates, D., & Paquette, S. (2011). Emergency knowledge management and social media technologies: A case study of the 2010 Haitian earthquake. *International Journal of Information Management*, *31*(1), 6–13.