Identifying Sub-events and Summarizing Disaster-Related Information from Microblogs

Koustav Rudra*
IIT Kharagpur, India
koustav.rudra@cse.iitkgp.ernet.in

Pawan Goyal
IIT Kharagpur, India
pawang@cse.iitkgp.ernet.in

Niloy Ganguly
IIT Kharagpur, India
niloy@cse.iitkgp.ernet.in

Prasenjit Mitra
The Pennsylvania State University,
University Park, PA, USA
pmitra@ist.psu.edu

Muhammad Imran
Qatar Computing Research Institute
(HBKU), Doha, Qatar
mimran@hbku.edu.qa

ABSTRACT

In recent times, humanitarian organizations increasingly rely on social media to search for information useful for disaster response. These organizations have varying information needs ranging from general situational awareness (i.e., to understand a bigger picture) to focused information needs e.g., about infrastructure damage, urgent needs of affected people. This research proposes a novel approach to help crisis responders fulfill their information needs at different levels of granularities. Specifically, the proposed approach presents simple algorithms to identify sub-events and generate summaries of big volume of messages around those events using an Integer Linear Programming (ILP) technique. Extensive evaluation on a large set of real world Twitter dataset shows (a). our algorithm can identify important sub-events with high recall (b). the summarization scheme shows (6—30%) higher accuracy of our system compared to many other state-of-the-art techniques. The simplicity of the algorithms ensures that the entire task is done in real time which is needed for practical deployment of the system.

CCS CONCEPTS

• **Information systems** → **Information retrieval diversity**; **Information extraction**; **Clustering and classification**; **Summarization**; *Web and social media search*;

KEYWORDS

Sub-event detection, humanitarian classes, class-based summarization, high-level summarization, situational information

1 INTRODUCTION

People at the scene of a disaster post information about the disaster on microblogs in real time. Emergency responses during natural or man-made disasters use information available on social media platforms such as Twitter. Volunteers and other support personnel generate reports or summaries of the relevant tweets posted via Twitter that the responders can then use to address the needs of people located in disaster areas [11]. Such manual intervention may not scale given the volume of data produced within a short time interval during a disaster. Moreover, different stakeholders and responders need information at varying levels of granularities. Some stakeholders may want to obtain overall situational updates for a given day as a short summary or report (**high-level information need**) or specific updates for a particular class, such as 'infrastructure damage', 'shelter needs' etc. (**class-specific information need**).

Several works on disaster-specific summarization [13, 17, 21, 22] in recent times proposed algorithms that mostly provide a general summary of the whole event. However, different stakeholders [29] like rescue workers, government agencies, field experts, common people, etc. have different information needs. Tweets posted during disasters contain information about various classes like 'infrastructure damage', 'missing or found people'. They can be classified using classification systems, e.g., AIDR [12]. To address the multi-dimensional needs of different stakeholders, we propose a *perspective-based tweet summarization* technique using an integer linear programming (ILP) framework. The framework provides flexibility to add constraints that capture the information needs of end-users (see Section 5).

Furthermore, existing works hardly leverage any disaster-specific trait to generate summaries. Every disaster witnesses a series of small-scale emergencies such as 'a bridge collapsing', 'airport getting shut', 'medical aid reaching an area', 'family members being stranded' etc - a summary on disaster may get enriched by including information of such sub-events. While interacting with experts at the United Nations Office for the Coordination of Humanitarian Affairs (OCHA), we realized that a summary can be enriched, if we include a representative but diverse sample of the important sub-events related to the disaster. Given the volume of the streaming data on a microblog like Twitter, the sub-event identification has to be fast (and thereby simple) but effective. We use a dependency parser to identify noun-verb pairs representing sub-events. Further, we rank them based on the frequency of co-occurrence of their constituent nouns and verbs in the corpus. Subsequent to the identification and ranking, maximization of high-ranked sub-events becomes one of the objectives of the linear program driving our tweet

*Koustav Rudra is presently affiliated to SONIC Lab, Northwestern University.

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07…$15.00
https://doi.org/10.1145/3209978.3210030

summarization framework. Our summarizer also utilizes multiple criteria where the constraints capture desirable properties of a good summary, and uses a fast ILP solver to generate summaries efficiently. The simplicity as well as the robustness of our approach makes it successful; we show empirically that despite being simple, this method works very well in practice.

We evaluate the efficacy of our (a) sub-event identification, and (b) tweet summarization algorithms against recently developed baselines [13, 17, 21, 22]. We evaluate our proposed methods on 1.87M, 0.49M, and 0.24M tweets collected using the AIDR platform [12] corresponding to the 2015 Nepal earthquake, the 2014 Typhoon Hagupit, and the 2014 Pakistan flood respectively using both traditional IR metrics and crowdsourcing. Our sub-event identification algorithm outperforms many state-of-the-art techniques [1, 2, 20, 30]. We find that elaborate clustering and dependency parsing based techniques perform poorly on 'noisy' tweet data.

Our proposed tweet summarization method performs 6-30% better in terms of ROUGE-1 F-score than existing methods. We crowdsource the evaluation of the quality of summaries and show that our method generates summaries that are rated to be significantly superior for post-disaster situation assessment compared to prior approaches [13, 17, 21, 22] in terms of information coverage, diversity, and readability (see Section 6). When we present our results, to increase readability, we highlight the sub-event keyphrases in the summary. An overwhelming majority of crowd-workers opined that highlighting helps them to grasp the situation summarized quickly. As a final contribution, we have made the codes and datasets publicly available at **http://www.cnergres.iitkgp.ac.in/subeventsummarizer/dataset.html**

2 RELATED WORK

Affected people and other observers post messages on microblogging sites such as Twitter in real-time during and after a disaster. Responders utilize these messages that provide situational information [5, 23, 27]. This information helps improve disaster relief operations [8, 11].

2.1 Sub-event detection during crises

Several studies have shown how to extract sub-events from tweets during disasters [1, 20, 26]. Recent approaches attempted to identify topics from evolving tweet streams [16, 24]. However, most existing topic/sub-event detection approaches cluster tweets using different approaches (self organizing map, latent dirichlet allocation, biterm topic modeling, nearest neighbour, etc.) and represent each cluster or some top frequency words from each cluster as topics or sub-events. End users find it difficult to understand a bag-of-words representing a sub-event. For example, traditional biterm topic models represent a topic using most probable terms like 'to', 'relief', 'Nepal', 'material', 'NDRF', etc. However, we believe that if we can identify sub-events like 'relief sent', 'NDRF rush', 'material carry', then it will be easier for end users to take decisions regarding relief and rescue operations. We try to go beyond existing clustering and bag-of-words based sub-event detection methods and provide a more user-friendly, understandable, and meaningful noun-verb pair based sub-event detection scheme, which is useful in disaster scenarios. For example, a sub-event like 'material carry' is easy to comprehend compared to a collection of words like 'plan, work, field, material' and we can easily understand from the first phrase that some relief material was dispatched to Nepal.

A recent short paper [2] on detecting sub-events from news articles uses sentential (similar to noun-verb) cues. Apart from that, the method depends on the dependency types like 'adjective modifier', 'adverbial complement' etc. present in a sentence; however, Stanford parser fails to identify such dependency types from noisy tweets with significant accuracy [9]. Furthermore, it starts with a small number of sentences and gradually adds new sentences having sub-events in each iteration. This process continues until the method converges and running time heavily depends on the convergence rate. Hence, it does not produce output in real-time over large datasets.

2.2 Tweet summarization

Several researchers have attempted to utilize information from Twitter to detect important situational updates from millions of posts on disaster-specific events [23, 28]. Methods for automatically generating summaries by extracting the most important tweets on the event have been proposed [17, 22]. Algorithms to summarize tweet streams in real-time have also been proposed recently [18, 24]. Osborne et al. proposed a real event-tracking system using greedy summarization [18]. Shou et al. [24] used clustering and LexRank [7] to generate extractive summaries from Twitter. Besides the above mentioned extractive summarization techniques, several abstractive methods have also been proposed. Rudra et al. [21] proposed a real-time abstractive tweet summarization technique suitable for disaster events. Kedzie et al. proposed an extractive summarization method to summarize disaster event-specific information from news articles [13].

Our contributions are based on two observations. First, a disaster comprises of various related sub-events. End-users can get improved situational awareness by reading summaries that contain mentions of important, representative sub-events than summaries that do not contain them. Second, a general summary is not able to satisfy the needs of different stakeholders like government, NGOs, rescue agencies, etc. In this paper, we propose a framework that can be used to generate summaries to satisfy information need at various granularities and different stakeholders.

3 DATASET AND HUMANITARIAN CATEGORY IDENTIFICATION

We collected crisis-related messages using the AIDR platform [12] from Twitter posted during three major disaster events — **(1) Nepal Earthquake (NEQuake):** This dataset consists of 1.87 million messages posted between April 25th and April 27th, 2015 fetched from Twitter using different keywords (e.g., Nepal Earthquake, NepalQuake, NepalQuakeRelief etc.). **(2) Typhoon Hagupit/Ruby (Hagupit):** This dataset consists of 0.49 million messages posted between December 6 and December 8, 2014 downloaded using different keywords (e.g., TyphoonHagupit, TyphoonRuby, Hagupit, etc.). **(3) Pakistan Flood (PFlood):** This dataset consists of 0.24M messages posted on September 7th and

Table 1: Description of the datasets corresponding to three different events. NA indicates the absence of a particular category for an event (i.e. no labeled data or the class contains very few tweets ($\leq$ 500)).

Category	NEQuake	Hagupit	PFlood
Missing, trapped, or found people	10,751	NA	2797
Infrastructure and utilities	16,842	3517	1028
Donation or volunteering services	1,530	4504	27,556
Shelter and supplies	19,006	NA	NA
Caution and advice	NA	25,838	NA
Displaced people and evacuations	NA	18,726	NA

Table 2: Popular sub-events learned from the first day of the Nepal earthquake (Apr 25, 2015).

Class	Sub-events
Infrastructure	'service affect', 'airport shut', 'road crack', 'building collapse'
Missing	'family stuck', 'tourist strand', 'database track'
Shelter	'field clean', 'medicine carry', 'emergency declare'

8th, 2014 obtained using different keywords (e.g., pakistanflood, PakistanFlood, Pakistanflood, etc.).

The datasets are classified into broad humanitarian categories using the AIDR [12] framework. These humanitarian categories are specified by humanitarian organizations such as UNOCHA and UNICEF based on their information needs. These classes may not remain the same across various disasters [6]. Table 1 shows the categories and detailed data statistics of three disaster events.

4 SUB-EVENT DETECTION - DEPSUB

Information about a humanitarian class (e.g., infrastructure damage) can be categorized into sub-events like 'airport shut', 'building collapse', etc. Such sub-events can be generated using Latent Dirichlet Allocation (LDA) [3], which outputs the most probable words belonging to each topic. However, domain experts at the United Nations Office for the Coordination of Humanitarian Affairs (OCHA) have found that LDA-based topics are too general to act upon [29].

Upon analyzing a few hundred tweets from each class and events' time-lines from web sources[1], we find that messages that report the most important sub-events after a major disaster, consist of two nuggets: 1) entity/noun (e.g. person, place, organization, etc.), i.e., the entity that the event is about, and 2) an action-part/verb (e.g. destroyed, closure, etc.), i.e., the part that specifies the type of incident that happened to the reported entity.

Table 2 provides examples of some sub-events from various classes. These sub-events show important yet very specific information after the Nepal earthquake disaster. The noun-verb pair may repeat in various context like 'family stuck' in Kathmandu; 'family stuck' in Pokran. We seek to generate these automatically.

Identifying Noun-Verb pairs: We extract nouns and verbs present in each message by using Twitter POS tagger [9]. However, detecting correct associations between nouns and verbs is a non-trivial task. For example, in the tweet: `#China media says buildings toppled in #Tibet _URL_`, both the words, 'says' and 'toppled' were identified as action verbs. The noun 'building' is related to the term 'toppled' but it is not related to the verb 'says'. Hence, ('building', 'toppled') forms a valid sub-event whereas ('building', 'says') does not. Note that, sometimes such nouns may not always appear prior or adjacent to the verbs in a tweet. For example, in the tweet: `India sent 4 Ton relief material, Team of doctors to Nepal`, ('material', 'sent') is a valid sub-event but the noun 'ton' appears closer to the verb 'sent' than the noun 'material'. Earlier, Cai et al. [4] showed dependency grammar based subject verb evaluation in formal sentences. Following their approach, we associate a noun to a verb accurately using the dependency edge information as obtained from the Twitter dependency parser [14].

Ranking sub-events: Using the above-mentioned approach, the number of automatically identified sub-events can be quite large. For this reason, we rank the identified sub-events based on different factors as described next. Since, a sub-event is represented by a noun-verb pair (e.g., ('tourist' 'stranded')), we postulate that a sub-event is important if the constituent words in the pair have not (or rarely) occurred separately in the document which means the noun-verb pair together covering different contexts like 'tourist stranded' in various places. Accordingly, we compute the Szymkiewicz-Simpson overlap score of a sub-event $S\ (N, V)$ using Equation 1:

$$Score(S) = \frac{|X \cap Y|}{min(|X|, |Y|)} \tag{1}$$

where X, Y indicate the set of tweets containing N and V, respectively.

However, Equation 1 does not discriminate between frequent and infrequent sub-events. To overcome this problem, we apply a discounting factor δ proposed by Pantel and Lin [19] to Equation 1. The discounting factor reduces the score of infrequent events.

$$\delta(S) = \frac{|X \cap Y|}{1 + |X \cap Y|} * \frac{min(|X|, |Y|)}{1 + min(|X|, |Y|)} \tag{2}$$

The weight of a sub-event S is computed as follows:

$$Weight(S) = Score(S) * \delta(S) \tag{3}$$

Our system ranks the sub-events based on their weights.

We refer to our DEpendency-Parser-based SUB-event detection approach comprising of identifying and ranking sub-events as **DEPSUB**. We evaluate the performance of DEPSUB in Section 6.

5 SUMMARIZATION ALGORITHM - SCC

Different humanitarian organizations operating during disasters have different information needs. Summarization frameworks should be able to cater to such varied needs. For example, some may only be interested in class-specific summaries while others interested in a high-level summaries of the situation. A good summary of a disaster situation should contain the following three ingredients: (a) information about the humanitarian class of interest to the end-user, (b) presence of highly ranked sub-events corresponding to each class, and, (c) 'maximum' occurrence of nouns, verbs, and numerals. The third criterion is taken from prior works, which show that in a disaster scenario, an effective summary can be generated by maximizing the number of content words [17, 22]. These three factors are put together in an ILP framework to summarize a set of tweets as discussed below. The importance of a content word is computed using the standard

[1] goo.gl/mKfdiy

Table 3: Notations used in the summarization technique.

Notation	Meaning
L	Desired summary length (number of words)
n	Number of *tweets* considered for summarization (in the time window specified by user)
m, p	Number of distinct content words and sub-events included in the n *tweets* respectively
q	Number of *classes* considered for summarization (each of the tweets belongs to some class)
i, j, k, a	index for *tweets, content words, sub-events, classes*
x_i	indicator variable for *tweet* i (1 if *tweet* i should be included in summary, 0 otherwise)
y_j	indicator variable for content word j
z_k	indicator variable for sub-event k
$Length(i)$	number of words present in *tweet* i
$Con\_Score(j)$	tf-idf score of content word j
$Sub\_Score(k)$	score of sub-event k
$ICL(a)$	importance/informative score of class a
TC_j	set of *tweets* where content word j is present
C_i, S_i	set of content words and sub-events present in *tweet* i
TS_k	set of *tweets* where sub-event k is present
$CL(i)$	class of tweet i
TCL_a	set of *tweets* belonging to *class* a
λ_1, λ_2	tuning parameter – relative weight for tweet, content word, and sub-event score

tf-idf score (with sub-linear tf scaling). Similarly, the weight of each sub-event is computed using Equation 3. We refer to our summarization framework as Sub-event-based Category-specific Content words summarization (SCC).

ILP Formulation: The summarization of L words is achieved by optimizing the following ILP objective function, whereby the highest scoring *tweets* are returned as the output of summarization. We use the GUROBI Optimizer [10] to solve the ILP. After solving this ILP, the set of *tweets* i such that $x_i = 1$, constitutes the summary. The symbols used in the following equations are as explained in Table 3.

$$max((1-\lambda_1-\lambda_2).\sum_{i=1}^{n} x_i.ICL(CL(i)) + \lambda_1.\sum_{j=1}^{m} Con\_Score(j).y_j.max_{i\in TC_j}(ICL(CL(i))) + \lambda_2.\sum_{k=1}^{p} Sub\_Score(k).z_k.max_{i\in TS_k}(ICL(CL(i)))) \quad (4)$$

In Equation 4, the scores of each of the content words and sub-events (suitably normalized in the [0,1] scale) are multiplied by the weight of the highest informative class in which this content word or sub-event is present (if the content word belongs to multiple classes). For example, (say) 'airport' belongs to both infrastructure and shelter classes; the weight of 'airport' is multiplied by the greater of the informative score of the 'infrastructure' and the 'shelter' classes. The importance of tweets, content-words, and sub-events is regulated by the parameters λ_1, λ_2.

The equation is, however, subject to the following constraints (Eqns. 5 - 11) which are explained below.

$$\sum_{i=1}^{n} x_i \cdot Length(i) \leq L \quad (5)$$

Eqn. 5 ensures that the total number of words contained in the *tweets* that get included in the summary is at most the desired length L (user-specified).

$$\sum_{i\in TC_j} x_i \geq y_j, j = [1\cdots m] \quad (6)$$

$$\sum_{j\in C_i} y_j = |C_i| \times x_i, i = [1\cdots n] \quad (7)$$

Eqn. 6 ensures that if the content word j is selected to be included in the summary, i.e., if $y_j = 1$, then at least one *tweet* in which this content word is present is selected. Eqn. 7 ensures that if a particular *tweet* is selected to be included in the summary, then the content words in that *tweet* are also included in summary.

$$\sum_{i\in TS_k} x_i \geq z_k, k = [1\cdots p] \quad (8)$$

$$\sum_{k\in S_i} z_k = |S_i| \times x_i, i = [1\cdots n] \quad (9)$$

Eqn. 8 ensures that if sub-event k is selected in the final summary i.e., if $z_k = 1$, then at least one *tweet* that covers that sub-event is selected. Eqn. 9 ensures that if a particular *tweet* is selected to be included in the summary, then the sub-events in that *tweet* are also considered in the summary.

$$\sum_{a=1}^{q} ICL(a) = 1, a = [1\cdots q] \quad (10)$$

Eqn. 10 ensures that sum of weight/ importance of all the classes is 1.

$$\sum_{i\in TCL_a} x_i \geq \delta, a = [1\cdots q] \text{ if } ICL_a > 0 \quad (11)$$

Eqn. 11 ensures that at least δ tweets from each class whose importance is greater than 0 will be included in the final summary.

Scenario Specific Summarization: As mentioned, the summarization can span across classes (high-level summary) or it can pertain to a particular class. The constraints set in the generalized equation can be customized for each case.
High-level summary: Given q classes, the importance of each class (ICL) is set to $\frac{1}{q}$ and the parameter δ in Eqn. 11 is set to 2, which represents the minimum number of tweets from each class that must be included in the final summary.
Class based summary: Given q classes, importance of all the classes, except the class for which a summary needs to be generated, is set to 0. It is set to 1 for the desired class.

6 EXPERIMENTAL SETUP AND RESULTS

The success of the summarization scheme (SCC) depends on efficient and accurate identification of the underlying sub-events (DEPSUB) (detail explained later using Table 11). Hence, first, we analyze the performance of DEPSUB and then SCC over datasets gathered from three recent disaster events - the Nepal Earthquake (NEQuake), the Typhoon Hagupit/Ruby (Hagupit) and the Pakistan Flood (PFlood). Details of these datasets are available in

Table 1. For NEQuake and Hagupit, there are data available for four different classes, and for PFlood, there are three. We further divided the tweets by date: 25th April to 27th April, 2015 for NEQuake, 6th December to 8th December, 2014 for Hagupit, and 7th September to 8th September, 2014 for the PFlood. For NEQuake and Hagupit, we created $4(class) \times 3(day) = 12$ instances. Similarly for PFlood, we created $3 \times 2 = 6$ instances. In this study, we have altogether 30 different instances/tasks[2].

6.1 Evaluation of DEPSUB

We evaluate the automatically identified sub-events to check their coverage with real-world events, their accuracy, and quality.

Baseline approaches: We use the following five state-of-the-art disaster specific sub-event detection approaches as our baselines. Among the baselines, the first method is an *NLP*-based technique whereas the next four are based on the clustering of related tweets.

(a). **Two-phase approach (TWS):** This is a dependency relation based sub-event detection approach applicable to news articles [2]. In order to make it suitable for tweets, we make the following modifications — (i) we remove Twitter specific tags such as hashtags, mentions, URLs, emoticons using Twitter POS tagger [9] from each tweet to provide clean data to the algorithm, (ii) their proposed method identifies sub-events from sentences that contain three or more verb phrases. However, we observe that hardly any tweet satisfies this constraint. Hence, we remove this constraint for tweets. TWS outputs ranked sub-events using dependency relations and the number of words in the sub-events vary based on the types of relations. For example, when a 'verb + direct object' is extracted, the sub-event is represented by two words, whereas when a 'verb + prepositional phrase' is extracted, the sub-event may contain more than two words.

(b). **COS-clustering:** Dhekar et al. [1] proposed a clustering based sub-event detection approach. We discard URLs, mentions, hashtag signs, emoticons, punctuation, and other Twitter-specific tags using a Twitter POS tagger [9]. Finally, in the labeling phase each sub-event cluster is represented by top four words having the highest term frequency among all the words belonging to that cluster.

(c). **LDA-clustering:** Blei et al. [3] proposed an LDA based topic modeling approach. Each sub-event/topic cluster is represented by top four words having the highest probability among all the words belonging to that topic cluster.

(d). **BTM-clustering:** Biterm topic modeling [30], similar to LDA-based clustering, was designed for short and informal texts like tweets, is used.

(e). **SOM-clustering:** Pohl, et al. [20] presents a self-organizing map-based automatic sub-events detection approach that we use as a baseline.

Evaluation methodology: To evaluate the importance and utility of sub-events identified by DEPSUB, we perform three different tests. We perform two user studies using (a) CrowdFlower[3] workers, and, (b) researchers having background knowledge about disasters (knowledgeable workers). Further, we evaluate the richness of the sub-events detected by comparing it against a prepared ground truth data.

[2]We have used terms 'instances' and 'tasks' interchangeably throughout this section.
[3]**http://www.crowdflower.com/**

Table 4: Results of the crowdsourcing based evaluation of sub-events for DEPSUB and baselines.

Datasets	Method	Evaluation					
		(a). Fraction of instances where a method wins			(b). Fraction of users voted for a method		
		Q1	Q2	Q3	Q1	Q2	Q3
NEQuake	DEPSUB	**0.58**	**0.68**	**0.75**	**0.43**	**0.38**	**0.38**
	TWS	0	0	0.25	0.17	0.07	0.23
	COS	0.17	0.08	0	0.13	0.23	0.12
	LDA	0.08	0.08	0	0.08	0.13	0.07
	BTM	0	0.08	0	0.05	0.15	0.07
	SOM	0.17	0.08	0	0.13	0.03	0.13
Hagupit	DEPSUB	**0.42**	**0.75**	**0.50**	**0.28**	**0.42**	**0.28**
	TWS	0.17	0.08	0.17	0.23	0.25	0.17
	COS	0.08	0	0.08	0.12	0.03	0.12
	LDA	0.16	0.08	0	0.13	0.15	0.12
	BTM	0	0.08	0.08	0.07	0.10	0.13
	SOM	0.17	0	0.17	0.17	0.05	0.18
PFlood	DEPSUB	**1**	**0.83**	**0.50**	**0.40**	**0.50**	**0.37**
	TWS	0	0.17	0.50	0.23	0.17	0.30
	COS	0	0	0	0.10	0.20	0.10
	LDA	0	0	0	0.10	0.07	0.07
	BTM	0	0	0	0.13	0.06	0.03
	SOM	0	0	0	0.03	0	0.13

(a). Evaluation using CrowdFlower workers: A CrowdFlower worker is provided with six lists of fifteen sub-events, as generated by six competing methods. For DEPSUB, we consider top fifteen sub-events based on the rank obtained via Eqn. 3. For TWS, we similarly collect top fifteen based on their frequency of occurrence in the corpus. In case of COS and SOM clustering method, clusters are ranked based on the number of tweets contained (highest ranked cluster contains the maximum number of tweets) and we select four words from each of the top fifteen clusters. Finally, for LDA and BTM, we set the number of topics to be fifteen and select four representative words with highest probability from each topic. The crowd-worker is asked to look at these lists, and answer the three questions elaborated below. Workers have to identify only the best of the six lists. That way the cognitive load on them is less. The order of the lists when presented is randomized to avoid any position bias that may influence the workers. We ask the following three questions:

(Q1) Which of the six methods identifies the least number of irrelevant sub-events? **(Q2)** Which of the six methods identifies sub-events most useful for crisis responders to understand the situation in the disaster region? **(Q3)** Which of the six methods is able to provide a clear situational overview (through the identified sub-events) of the disaster situation stated above?

We have 30 instances from the three events; each of these instances are assessed by 15 crowd-workers. That means $30 \times 15 = 450$ (180, 180, 90) votes were obtained altogether. The outcome of the survey is represented in Table 4. Questions Q1, Q2, Q3 are subjective in nature and we obtain fair (0.22) Fleiss-Kappa agreement score among the annotators. For each task, the technique (among six) that gets the most votes among the 15 answers is chosen as the winner. In this manner statistics showing the fraction of times a technique has won is reported in columns [3-5] of Table 4. For each question, we report the fraction of user votes received by a particular method in columns [6-8] of Table 4. From the table we find that DEPSUB is a winner in all the three aspects probed - the margins with others are particular high in the cases of Q1 and Q2. That implies the

crowd-workers have found DEPSUB generated sub-events least irrelevant and most useful. In case of providing situation updates, TWS, although second, performs well. These findings also establish the importance of *nlp* based techniques such as DEPSUB and TWS in identifying sub-events to capture situational information.

(b) Evaluation by knowledgeable users: Since we could not assume any background knowledge of the CrowdFlower workers, we further decided to conduct a small scale experiment with knowledgeable workers to cross-check the accuracy of the results and also to rank the techniques (ranking would have been too much of a cognitive overload for the crowd-workers). The current work is a part of a multi-institutional project on post-disaster management involving around 30 graduate students[4] - 20 of them (15 male and 5 female candidates) took part in this survey.

From Table 4, we can observe that user preferences are distributed across DEPSUB, TWS, COS, and SOM methods which are specifically designed to extract sub-events during crisis. Hence, for this ranking task, we only considered these four methods and asked them to rank the methods based on the above mentioned three questions (Q1, Q2, and Q3). Every question in an instance received 20 different answers i.e., altogether $30(\#instances) \times 20(\#users) = 600$ responses. Table 6 shows a summary matrix [25] outlining the performance of DEPSUB i.e., fraction of responses preferring DEPSUB over three competitive baselines. Their voting patterns also roughly matched with that of the crowd-workers (results not shown). From Table 6, we observe several interesting results — (a) DEPSUB almost always is able to maintain its rank within the first two positions in the list for all the three questions. (b) TWS is a clear second and its performance is particularly commendable with respect to (Q2) (Q3), this result is a bit different from what we received from the crowd-workers where TWS performance w.r.t Q2 was much inferior. (d). We found TWS performance deteriorates in the sets where there are a sizeable fraction (detail results omitted due to paucity of space) of *more than two words* sub-events.

(c). Evaluation using gold standard sub-events:
We create a ground truth of sub-events and compare the efficiency of DEPSUB using standard IR metrics precision, recall, and F-score by the amount of overlap it has with the gold standard. We explain how the gold standard is generated.

Establishing gold standard sub-events: Three human volunteers individually prepared sub-events as pairs of nouns and verbs for each of the instances (12 for both NEQuake, Hagupit and 6 for PFlood). To prepare the final gold standard sub-events for a particular instance, we chose those sub-events that were selected by at least two volunteers. For comparison, we only consider TWS because its output format is quite similar to DEPSUB. While TWS produces a fraction of sub-events containing more than two words, experiment results showed (mentioned in previous part) that most of them are of poor quality, so we only chose only two worded sub-events from TWS to initiate a fair comparison[5].

Table 5 shows the precision, recall, and F-scores for DEPSUB and TWS for the three datasets, over various days and classes respectively. From Table 5, we can see that DEPSUB performs on an average around 30% better than TWS in terms of precision, recall and F-score. The accuracy of TWS is dependent on the accurate identification of the verb phrases and its dependents. The Stanford parser does not work well in extracting verb phrases and named dependencies (e.g., 'amod', 'ccomp' etc.) from Twitter. On the other hand, DEPSUB represents sub-events simply as a pair of action (verb) and subject of that action (noun). In DEPSUB we observe that some of the rarely/infrequently occurring noun-verb pairs ('afternoon fly', 'terminal flee') do not make any sense. For noun-verb association, we rely on a Twitter dependency parser [14], which has its own limitations due to the noisy and informal nature of tweets. These shortcomings hamper the precision of the system by $\approx 30\%$.

6.2 Evaluation of SCC

We compare the performance of SCC with the recent disaster-specific and real-time summarization approaches. We discuss the baseline techniques and the experimental settings briefly, and then compare the performance of the techniques. Disaster response planners use class-specific summaries instead of a high-level summary [6, 11]. We evaluate the performance of SCC with respect to its ability to produce class-specific summarizations and then check the quality of the high-level summaries. Besides comparing it with the baseline algorithms, we also check the utility of including sub-events in the summaries as well as that of the content word component in the summarization framework. Furthermore, we check the utility of putting a soft (minimum) constraint on the number of tweets to be present from each class (Eqn. 11) in a high-level summary.

Establishing gold standard summaries: Three volunteers (none of them is an author of this paper) individually prepared summaries of length 200 words from the tweets for each of the 30 instances of data we used for evaluating *class-based summarization*. To prepare the final gold standard summary for a particular instance, first, we chose those tweets that were included in the individual summaries written by all the volunteers, followed by those tweets that were included by a majority of the volunteers until we reach the word limit. Thus, we create a gold-standard summary containing 200 words for each instance. Similarly, to test the quality of the *high-level summaries* generated by our system, volunteers produce summaries for each day for each of the three datasets.

Baseline approaches: We use the following four state-of-the-art disaster specific summarization approaches as our baselines:

(1) **COWTS:** is an extractive summarization approach specifically designed for generating summaries from disaster-related tweets [22].
(2) **COWABS:** an abstractive disaster-specific summarization approach proposed by Rudra et al. [21].
(3) **APSAL:** is an affinity clustering based extractive summarization technique proposed by Kedzie et al. [13]. It is designed for news articles; hence, we apply APSAL over tweets after removing Twitter specific tags like URLs, hashtags, mentions, emoticons etc. using the POS tagger [9].
(4) **TSum4act:** is a disaster-specific summarization approach proposed by Nguyen et al. [17].

[4]Detail omitted to maintain anonymity
[5]For this analysis, we consider all the two-worded sub-events generated by DEPSUB and TWS.

Table 5: Precision (P), Recall (R), F-scores (F) over different datasets for DEPSUB and TWS for sub-event detection.

Event	Date	Infrastructure						Missing						Shelter						Volunteer					
		DEPSUB			TWS			DEPSUB			TWS			DEPSUB			TWS			DEPSUB			TWS		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
NEQuake	25/04/2015	0.57	0.91	0.70	0.35	0.60	0.44	0.72	0.79	0.75	0.57	0.62	0.59	0.85	0.96	0.90	0.62	0.81	0.70	0.41	0.71	0.52	0.34	0.40	0.37
	26/04/2015	0.52	0.88	0.66	0.27	0.55	0.36	0.83	0.88	0.85	0.64	0.85	0.73	0.78	0.96	0.87	0.54	0.92	0.68	0.71	0.84	0.77	0.42	0.54	0.47
	27/04/2015	0.46	0.83	0.60	0.35	0.54	0.43	0.81	0.95	0.87	0.61	0.71	0.66	0.69	0.98	0.81	0.53	0.86	0.65	0.77	0.83	0.80	0.37	0.25	0.30

Event	Date	Infrastructure						Caution						Displaced						Volunteer					
		DEPSUB			TWS			DEPSUB			TWS			DEPSUB			TWS			DEPSUB			TWS		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Hagupit	06/12/2014	0.80	0.85	0.83	0.52	0.80	0.63	0.67	0.83	0.74	0.42	0.83	0.56	0.57	0.83	0.68	0.27	0.73	0.39	0.83	0.85	0.84	0.48	0.74	0.58
	07/12/2014	0.81	0.92	0.86	0.63	0.76	0.69	0.65	0.88	0.75	0.46	0.82	0.59	0.54	0.85	0.66	0.26	0.78	0.40	0.68	0.86	0.76	0.52	0.92	0.67
	08/12/2014	0.65	0.67	0.66	0.39	0.60	0.47	0.68	0.91	0.77	0.49	0.79	0.60	0.63	0.85	0.72	0.33	0.42	0.37	0.76	0.86	0.81	0.59	0.69	0.64

Event	Date	Infrastructure						Missing						Volunteer					
		DEPSUB			TWS			DEPSUB			TWS			DEPSUB			TWS		
		P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
PFlood	07/09/2014	0.60	0.76	0.67	0.35	0.46	0.40	0.71	0.93	0.80	0.30	0.35	0.32	0.68	0.95	0.82	0.48	0.92	0.63
	08/09/2014	0.78	0.74	0.76	0.23	0.40	0.30	0.75	0.90	0.82	0.50	0.46	0.48	0.67	0.93	0.80	0.50	0.90	0.64

Table 6: Fraction of responses that prefer DEPSUB over other baselines.

Datasets	#Responses	Fraction of responses that prefer DEPSUB								
		Q1			Q2			Q3		
		Over TWS	Over COS	Over SOM	Over TWS	Over COS	Over SOM	Over TWS	Over COS	Over SOM
NEQuake	240	0.68	0.87	0.93	0.69	0.83	0.97	0.65	0.80	0.95
Hagupit	240	0.75	0.90	1	0.83	0.82	0.97	0.60	0.97	1
PFlood	120	0.85	0.92	1	0.77	0.94	1	0.57	0.89	0.89

Evaluation metrics: We perform two types of evaluations. First, we use the standard ROUGE [15] metric for evaluating the quality of summaries generated using the proposed as well as the baseline methods. In this case, due to the informal nature of tweets, we consider the recall and F-score of the ROUGE-1 variant only. Second, we perform user studies using CrowdFlower. For all cases, we generated a system summary of 200 words for SCC and for each of the baselines over each instance. For SCC, we have tried different values for λ_1, and λ_2. Based on the ROUGE-1 score, we determine the best values for λ_1, and λ_2. λ_1 = 0.5, and λ_2 = 0.3 are the best weights for Hagupit and PFlood. Weights for the NEQuake are λ_1 = 0.5, and λ_2 = 0.5.

6.2.1 Performance evaluation of class based summarization. The format in which the summary is produced is highlighted in Table 9. We produce extractive summary by choosing the tweets which have been selected through the ILP framework. Besides that the sub-event (if) responsible for selection of a tweet is highlighted. To judge the quality of SCC, we perform a CrowdFlower based qualitative analysis along with ROUGE based quantitative judgement.

Evaluation using gold-summaries: Table 7 shows the ROUGE-1 F-scores for the five algorithms for 24 instances[6] spanning all the three datasets. SCC performs significantly better than all the baseline approaches. For instance, mean scores indicate an average improvement of more than 6%, 24%, and 30% respectively with respect to F-scores in comparison to extractive summarization schemes COWTS [22], APSAL [13], and TSum4act [17] and 20% over abstractive summarization technique COWABS [21].

Evaluation using crowdsourcing: We used the CrowdFlower crowdsourcing platform to evaluate summaries generated by SCC and all the four baselines. In total, we have 12 instances (hence 60 summaries) for the NEQuake and Hagupit and 6 instances (hence 30 summaries) for the PFlood. A crowdsourcing task, in this case, consists of five summaries and the four evaluation criteria (as described below). Each task requires ten different workers' response on an answer before we finalize it. Table 8 summarizes the result of the crowdsourcing evaluation on a per event basis. Note there are 120 respondents for NEQuake and Hagupit corresponding to each question while this is 60 for PFlood. The exact description of the crowdsourcing task is as follows:

"The purpose of this task is to evaluate machine-generated summaries using tweets collected during the Nepal Earthquake of 2015, and, the Typhoon Hagupit and the Pakistan Flood, both in 2014. We aim to built an automatic method to generate such summaries/reports useful for situational awareness (information that helps understand the situation on the ground after an incident) to crisis responders. For this purpose, we have used five different methods and we want to compare which one is better. Given the summaries and their topic, we are interested in comparing them based on the following criteria: Information coverage, Diversity and Readability.

Given the summaries and their topic, We asked four questions to the workers on CrowdFlower as follows — **(Q1)** Overall, which method in your opinion has the best information coverage? **(Q2)** Overall, which method covers most diverse topics? **(Q3)** Overall, which summary helps you quickly understand and comprehend the situation? **(Q4)** Overall, do you prefer summaries with highlighted topics or without highlighting?

Q1. Information coverage corresponds to the richness of information a summary contains. For instance, a summary with more informative sentences (i.e., crisis-related information) is considered better in terms of information coverage. From Table 8, we can see that SCC is able to capture more informative summary than other baseline approaches in around 50% of the cases. The performance of the competing algorithms varies from event to event, so we don't see any of the techniques consistently coming second best across all events.

Q2. Diversity of topics tries to capture variation of information in a summary. While we do not use any explicit parameter to control diversity, our proposed ILP framework captures diverse set of information through the selection of various content words and sub-events. Eqn. 4 considers the weight of each of the content words

[6]Similar trend is observed for rest of the six instances from volunteer class of NEQuake and Hagupit.

Table 7: Comparison of ROUGE-1 F-scores for SCC and the four baseline methods (COWTS, COWABS, APSAL, and TSum4act) on the same situational tweet stream for each class, for each day, and for each dataset.

Date	ROUGE-1 F-score (NEQuake)														
	Infrastructure					Missing					Shelter				
	SCC	COWTS	COWABS	APSAL	TSum4act	SCC	COWTS	COWABS	APSAL	TSum4act	SCC	COWTS	COWABS	APSAL	TSum4act
25/04/2015	**0.4966**	0.4842	0.3866	0.3691	0.3758	**0.5407**	0.5353	0.3082	0.3162	0.1901	**0.5503**	0.5165	0.4548	0.4513	0.4742
26/04/2015	**0.3719**	0.3496	0.3496	0.3071	0.2387	**0.3848**	0.3066	0.3034	0.3496	0.3694	**0.3689**	0.3674	0.3387	0.3275	0.3610
27/04/2015	**0.4971**	0.3631	0.3352	0.3657	0.3765	**0.3574**	0.3494	0.3275	0.3478	0.2825	**0.4573**	0.4340	0.3922	0.3238	0.3631
Date	**ROUGE-1 F-score (Hagupit)**														
	Infrastructure					Caution					Displaced				
	SCC	COWTS	COWABS	APSAL	TSum4act	SCC	COWTS	COWABS	APSAL	TSum4act	SCC	COWTS	COWABS	APSAL	TSum4act
06/12/2014	**0.6200**	0.6190	0.5364	0.4946	0.5655	**0.4658**	0.4498	0.4259	0.2922	0.3566	**0.3989**	0.3955	0.3676	0.2881	0.2558
07/12/2014	**0.6177**	0.6173	0.4702	0.4339	0.4852	**0.3363**	0.3303	0.3333	0.3202	0.3281	**0.3718**	0.3585	0.2905	0.2500	0.2307
08/12/2014	**0.4857**	**0.4857**	0.4637	0.3891	0.4413	**0.4175**	0.4169	0.3147	0.3803	0.4125	**0.4277**	**0.4277**	0.4144	0.3376	0.3812
Date	**ROUGE-1 F-score (PFlood)**														
	Infrastructure					Missing					Volunteer				
	SCC	COWTS	COWABS	APSAL	TSum4act	SCC	COWTS	COWABS	APSAL	TSum4act	SCC	COWTS	COWABS	APSAL	TSum4act
07/09/2014	**0.7306**	0.7232	0.6762	0.6894	0.7191	**0.6039**	**0.6039**	0.5705	0.5787	0.5769	**0.3651**	0.3378	0.3459	0.2646	0.2092
08/09/2014	**0.7235**	0.7206	0.6926	0.6781	0.6315	**0.4758**	**0.4758**	0.4436	0.4705	0.4498	**0.3844**	0.2865	0.3227	0.2105	0.2631

Table 8: Results of the crowdsourcing-based evaluation of class-based summaries generated by SCC and the four baseline techniques. Values in the table indicate percentage(%) of times a method is preferred for a particular question (NA indicates question is not valid for a method).

Datasets	Method	Q1	Q2	Q3	Q4
NEQuake	SCC	0.42	0.33	0.58	0.83
	COWTS	0.25	0.17	0.17	NA
	COWABS	0.25	0.17	0.25	NA
	APSAL	0.08	0.25	0	NA
	TSum4act	0	8	0	NA
Hagupit	SCC	0.58	0.50	0.59	0.92
	COWTS	0	8	8	NA
	COWABS	0.25	0.42	0.25	NA
	APSAL	0.17	0	8	NA
	TSum4act	0	0	0	NA
PFlood	SCC	0.50	0.83	0.83	0.83
	COWTS	0.33	0.17	0.17	NA
	COWABS	0.17	0	0	NA
	APSAL	0	0	0	NA
	TSum4act	0	0	0	NA

and sub-events only once (y_j, z_k are binary indicators); hence, tries to cover more number of highly ranked content words, sub-events to maximize the objective function. Although SCC performs best in all the cases, its performance varies widely, from modest 33% in NEQuake to 83% in PFlood.

Q3. Summary understanding attempts to measure how easy it is to comprehend the summary. This is where we ask the workers whether they get a mental picture of the situation and can think of some action after reading the summary. From Table 8, we see that majority of respondents found that SCC facilitates quick understanding of the situation.

Q4. Necessity of sub-event highlight tries to measure whether users prefer highlighting and that helps in improving comprehension. In Table 8, we can see that a large majority of respondents (Nepal - 83%, Hagupit - 92%, Pakistan - 83%) found highlighting provide more vivid picture compared to the flat versions.

Table 9: Summary of length 50 words (excluding #,@,RT,URLs), generated from the situational tweets of the infrastructure class (26th April) by (i) SCC (proposed methodology), (ii) COWABS.

Summary by SCC	Summary by COWABS
All flights canceled as **airport closes** down after quake. Reporter kathmandu airport closed **following** 6.7 **aftershock** no planes allowed to land. Metropolitan police department **rescue team** at airport to nepal. Kathmandu **airport reopened**. Nepal quake photos show historic **buildings reduced** to rubble as survivor search continues. Death **toll** in the earthquake in nepal **exceeded** 2 thousand people	Update flight operation starts from tribhuvan international airport, kathmandu video. Aftermath 7.8 earthquake in nepal this time 6.8 (dies). Nepal quake photos show historic buildings reduced to rubble as survivor search continues. Nepal death toll nears 2,000 as major 6.7 aftershock strikes kathmandu and experts say disaster was a.

Table 10: Runtime (seconds) of different algorithms for each of the classes averaged over three days.

Datasets	Class	SCC	COWTS	COWABS	APSAL	TSum4act
NEQuake	infrastructure	130.17	12.88	21.56	1719.79	16.79K
	missing	103.96	7.20	21.24	646.18	7.97K
	shelter	226.70	16.78	29.51	2685.67	21.45K
	volunteer	22.58	1.98	9.66	10.35	0.84K
Hagupit	infrastructure	63.92	3.02	11.02	57.50	2.01K
	caution	205.97	19.91	28.15	3846.34	33.30K
	displaced	152.10	17.06	31.14	2144.39	22.22K
	volunteer	38.86	4.07	17.03	103.67	2.70K
PFlood	infrastructure	13.62	1.82	8.60	11.37	0.78K
	missing	42.32	3.61	18.44	100.13	2.55K
	volunteer	390.68	56.02	62.15	11542.43	75.69K

Table 9 shows summary snippets generated by SCC and COWABS (both disaster-specific methodologies) from the same set of messages (i.e., tweets from infrastructure class posted on 26th April). The two summaries are quite distinct. On manual inspection, we felt that summary returned by SCC is more informative and diverse in nature compared to COWABS. For instance, we can see the SCC summary contains information about flights, damages of buildings, closing and reopening of airport, etc.

Time taken for summarization: As stated earlier, one of our primary objectives is to generate the summaries in real-time. Hence, we analyze the execution times of the various techniques. Table 10

provides detailed information about run-time of our proposed SCC method[7] and four other baselines. APSAL requires more time over large datasets because it performs non-negative matrix factorization and, affinity clustering. Its running time increases exponentially with the number of tweets. TSum4act takes more time due to detection of optimal number of topics, application of PageRank algorithm over tweets etc. SCC has a higher running time compared to COWTS [22] and COWABS [21] - the time mainly is taken to identify sub-events. However, the algorithm still can be considered as *near real-time* as typically a summary would be produced (say) after every half an hour.

Discussion on performance: APSAL [13] maintains clusters of related information and finally chooses an exemplar tweet from each cluster. Importance of exemplar tweets are decided based on some features. However, this method is designed for formal news articles; hence, many of its features (position of a sentence in a document, an event-type-specific language model) are missing for the tweets which are noisy and informal in nature. TSum4act assumes uniform importance for each of the clusters and selects one tweet per cluster which may not be true during disaster scenario. Some clusters may be more important compared to others and selection of more than one tweet from those clusters may lead to better summarization output. Moreover, we observe one interesting phenomenon here – SCC outperforms both the extractive (COWTS) and the abstractive (COWABS) approaches. Primarily there are two reasons behind that: (i) in the path formation step, the bi-gram model is followed in the abstractive approach (COWABS), which limits the scope of path formation. Hence, some important informational tweets are dropped because such tweets can not be combined with any other related tweets. (ii) in our proposed method, we work with both the content words, and sub-events, and we have already shown in Section 4 that such sub-events are better representative of real-world events.

Effect of content words and sub-events: Since SCC optimizes both presence of content words and sub-events, we performed extensive experiments to dissect importance of each. Table 11 compares the F-scores (averaged over three days) obtained considering both sub-events and content words, with those obtained considering any one of these parameters. The results show that both content words and sub-events contribute to the quality of the summary, and removing either decreases the overall performance in all the cases.

6.2.2 Performance evaluation of high-level summary. The summary generated by SCC has the following components: (a) selected tweets, (b) highlights of sub-events and a mention of the class from which an individual tweet has been selected, and, (c) classwise distribution of tweets in the summary. For example, on 25th April in NEQuake event, the high-level summary contains 33% tweets from infrastructure, 13% from missing, 17% from shelter and 37% from volunteer classes respectively. Presentation of such meta-data along with the tweets helps in comprehension - which we have also got confirmed through crowdsourced experiment (detailed result omitted due to paucity of space). We present ROUGE based evaluation and comparison with baselines.

[7]For SCC we consider the time taken to generate the sub-events and producing the final summary.

Table 11: Effect of content words and sub-events on summarization

Datasets	Class	SCC	SCC(content)	SCC(sub-event)
NEQuake	infrastructure	**0.4552**	0.3989	0.4193
	missing	**0.4276**	0.3977	0.3404
	shelter	**0.4588**	0.4393	0.3958
	volunteer	**0.5730**	0.5578	0.5338
Hagupit	infrastructure	**0.5744**	0.5740	0.4385
	caution	**0.4065**	0.3990	0.3226
	displaced	**0.3994**	0.3946	0.3107
	volunteer	**0.4589**	0.4483	0.3917
PFlood	infrastructure	**0.7270**	0.7195	0.6112
	missing	**0.5260**	0.5250	0.5363
	volunteer	**0.3747**	0.3263	0.3742

Table 12: Comparison of ROUGE-1 F-scores for SCC and the four baseline methods on the same tweet stream for each dataset, for each day.

Datasets	Day	SCC	COWTS	COWABS	APSAL	TSum4act
NEQuake	25/04/2015	**0.4117**	0.3662	0.3413	0.2215	0.3241
	26/04/2015	**0.3055**	0.2896	0.2087	0.3055	0.2666
	27/04/2015	**0.3853**	0.3726	0.3353	0.2866	0.3087
Hagupit	06/12/2014	**0.3223**	0.3008	0.3204	0.1943	0.2460
	07/12/2014	**0.4124**	0.3569	0.2832	0.2314	0.2492
	08/12/2014	**0.3475**	0.3002	0.3315	0.2128	0.2359
PFlood	07/09/2014	**0.4524**	0.4141	0.3316	0.2016	0.3014
	08/09/2014	**0.4145**	0.3085	0.3575	0.1823	0.2030

Performance: Table 12 gives the ROUGE-1 F-scores for the five algorithms for the three datasets over different days. As expected, SCC performs the best but its performance over other baselines is much better than individual class-based case (Table 7). The extra improvement seems to be coming from its ability to select proportionately from various constituent classes which is enabled by the constraints defined through Eqns. 10 - 11. The baseline algorithms on the other hand had to be fed with the entire data set excluding class information as there is no mechanism for them to use that information.

Role of humanitarian classes: We further test the impact of the constraints in the quality of summary generation by considering three competing variant of SCC. (i) We create a summary for each class using SCC and then combine them to form the high-level summary of 200 words such that each class is represented by the same number of words in the high-level summary (SCC(uniform)), (ii) we measure the proportion of tweets present in different classes (each day) and distribute the 200 word limit across these classes based on that proportion (SCC(proportion)), and (iii). we remove the class based constraint (Eqn. 11) i.e., number of tweets (two) to be present from each class in the high-level summary from the ILP framework. In this case, we don't consider any class information and take whole set of tweets (SCC(whole)).

Table 13 shows the performance of SCC and its variations. The uniform word selection and constraint removed strategies perform poorly in comparison to SCC. This is because the importance of individual classes varies over the day and a summary needs to capture that. Summaries generated based on the proportion of tweets present in different classes are not able to beat SCC. There are broadly two reasons behind that — (i) lots of retweets and near duplicate tweets are present in each of these classes and number of tweets is not able to represent word proportions accurately,

Table 13: Comparison of ROUGE-1 F-scores for SCC (the proposed methodology) and its three variations on the same tweet stream for each dataset, for each day

Datasets	Day	SCC	SCC (uniform)	SCC (proportion)	SCC (whole)
NEQuake	25/04/2015	**0.4117**	0.2598	0.3058	0.3095
	26/04/2015	**0.3055**	0.3033	0.2758	0.2809
	27/04/2015	**0.3853**	0.3687	0.3613	0.3416
Hagupit	06/12/2014	**0.3223**	0.3108	0.3080	0.3176
	07/12/2014	**0.4124**	0.3172	0.3046	0.3064
	08/12/2014	**0.3475**	0.2849	0.2608	0.3475
PFlood	07/09/2014	**0.4524**	0.4173	0.3886	0.4365
	08/09/2014	**0.4145**	0.3197	0.3529	0.3621

(ii). some of the overlapping information is present in more than one class (e.g., information about airport, flight is present in both 'infrastructure' and 'shelter' class) and independent consideration of the classes fails to capture this phenomenon. Note that SCC can dynamically adjust the proportion of each class as per 'real' content and hence provides superior summaries.

7 CONCLUSION

After interacting with several responders, we realized that summarization of information in the tweets from various perspectives and producing a summary focusing on sub-events is a pressing need in the real world. Accordingly, we have proposed a simple summarization approach, which can generate summaries across various scenarios. Specifically, in this paper, we have considered summaries : (i) of the overall situation, and (ii) of different humanitarian classes. We proposed DEPSUB, a sub-event identification algorithm. A crowdsourced evaluation of DEPSUB showed it to be superior in terms of relevance, usefulness as well as expressiveness. Summaries generated by DEPSUB were rated to be in the top two among five competing algorithms; this observation was confirmed by a quantitative evaluation using ROUGE-1 scores. Our proposed summarization algorithm, SCC - was rated to be superior in terms of diversity, coverage and understandability. Highlighting of sub-events also made the summary more understandable. SCC outperformed baseline algorithms between 6-30%; specifically, we show that the improvement resulted from the inclusion of sub-events. The importance of the different humanitarian classes (infrastructure, missing, shelter etc.) varies over days. SCC nicely captures and adjusts to the changing need. To the best of our knowledge, our work is the first to propose a comprehensive multi-faceted summarization approach; the framework developed can be applied to several important specialized situations (e.g. summarizing missing people information, geography-centric information etc.) - some of which will be our immediate future work.

Acknowledgement: Funding for this project was in part provided by Army Research Lab grant W911NF-09-2-0053.

REFERENCES

[1] Dhekar Abhik and Durga Toshniwal. 2013. Sub-event detection during natural hazards using features of social media data. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 783–788.

[2] Allison Badgett and Ruihong Huang. 2016. Extracting Subevents via an Effective Two-phase Approach.. In *EMNLP*. 906–911.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[4] Dongfeng Cai, Yonghua Hu, Xuelei Miao, and Yan Song. 2009. Dependency Grammar Based English Subject-Verb Agreement Evaluation.. In *PACLIC*. Citeseer, 63–71.

[5] Mark A. Cameron, Robert Power, Bella Robinson, and Jie Yin. 2012. Emergency Situation Awareness from Twitter for Crisis Management. In *Proc. WWW*. ACM, 695–698.

[6] Carlos Castillo. 2016. *Big Crisis Data: Social Media in Disasters and Time-Critical Situations* (1st ed.). Cambridge University Press, New York, NY, USA.

[7] Gunes Erkan and Dragomir R. Radev. 2004. LexRank:Graph-based lexical centrality as salience in text summarization. *Artificial Intelligence Research* 22 (2004), 457–479.

[8] Huiji Gao, Geoffrey Barbier, and Rebecca Goolsby. 2011. Harnessing the Crowdsourcing Power of Social Media for Disaster Relief. *Intelligent Systems, IEEE* 26, 3 (2011), 10–14.

[9] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah Smith, A. 2011. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proc. ACL*.

[10] gurobi 2015. Gurobi – The overall fastest and best supported solver available. http://www.gurobi.com/.

[11] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing social media messages in mass emergency: a survey. *ACM Computing Surveys (CSUR)* 47, 4 (2015), 67.

[12] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. Aidr: Artificial intelligence for disaster response. In *Proc. WWW companion*. 159–162.

[13] Chris Kedzie, Kathleen McKeown, and Fernando Diaz. 2015. Predicting Salient Updates for Disaster Summarization. In *Proc. ACL*. Beijing, China, 1608–1617.

[14] Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A Dependency Parser for Tweets. In *Proc. EMNLP*.

[15] Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proc. Workshop on Text Summarization Branches Out (with ACL)*.

[16] Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. 2015. Degeneracy-based real-time sub-event detection in twitter stream. In *Proc. AAAI ICWSM*. 248–257.

[17] Minh-Tien Nguyen, Asanobu Kitamoto, and Tri-Thanh Nguyen. 2015. TSum4act: A Framework for Retrieving and Summarizing Actionable Tweets during a Disaster for Reaction. In *Proc. PAKDD*.

[18] Miles Osborne, Sean Moran, Richard McCreadie, Alexander Von Lunen, Martin Sykora, Elizabeth Cano, Neil Ireson, Craig Macdonald, Iadh Ounis, Yulan He, Tom Jackson, Fabio Ciravegna, and Ann OBrien. 2014. Real-Time Detection, Tracking, and Monitoring of Automatically Discovered Events in Social Media. In *Proc. ACL*.

[19] Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 613–619.

[20] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2012. Automatic sub-event detection in emergency management using social media. In *Proc. WWW*. ACM, 683–686.

[21] Koustav Rudra, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, and Prasenjit Mitra. 2016. Summarizing Situational Tweets in Crisis Scenario. In *Proceedings of the 27th ACM Conference on Hypertext and Social Media*. ACM, 137–147.

[22] Koustav Rudra, Subham Ghosh, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. 2015. Extracting Situational Information from Microblogs during Disaster Events: a Classification-Summarization Approach. In *Proc. CIKM*.

[23] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proc. WWW*. 851–860.

[24] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. 2013. Sumblr: Continuous Summarization of Evolving Tweet Streams. In *Proc. ACM SIGIR*. 533–542.

[25] summary-matrix 2017. Summary Matrix — Kemeny-Young method. **`https://en.wikipedia.org/wiki/Kemeny-Young_method`**.

[26] Lynda Tamine, Laure Soulier, Lamjed Ben Jabeur, Frederic Amblard, Chihab Hanachi, Gilles Hubert, and Camille Roth. 2016. Social Media-Based Collaborative Information Access: Analysis of Online Crisis-Related Twitter Conversations. In *ACM 27th Conference on Hypertext & Social Media*.

[27] Istvan Varga, Motoki Sano, Kentaro Torisawa, Chikara Hashimoto, Kiyonori Ohtake, Takao Kawai, Jong-Hoon Oh, and Stijn De Saeger. 2013. Aid is Out There: Looking for Help from Tweets during a Large Scale Disaster.. In *Proc. ACL*.

[28] Sudha Verma, Sarah Vieweg, William J. Corvey, Leysia Palen, James H. Martin, Martha Palmer, Aaron Schram, and Kenneth M. Anderson. 2011. Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency. In *Proc. AAAI ICWSM*.

[29] Sarah Vieweg, Carlos Castillo, and Muhammad Imran. 2014. Integrating social media communications into the rapid assessment of sudden onset disasters. In *Social Informatics*. Springer, 444–461.

[30] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *Proc. WWW*. ACM, 1445–1456.