Summarizing Situational Tweets in Crisis Scenario

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

Siddhartha Banerjee
The Pennsylvania State University, USA
sub253@ist.psu.edu

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

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

Muhammad Imran
Qatar Computing Research Institute, HBKU, Doha, Qatar
mimran@qf.org.qa

Prasenjit Mitra
Qatar Computing Research Institute, HBKU, Doha, Qatar
pmitra@qf.org.qa

ABSTRACT

During mass convergence events such as natural disasters, microblogging platforms like Twitter are widely used by affected people to post situational awareness messages. These crisis-related messages disperse among multiple categories like infrastructure damage, information about missing, injured, and dead people etc. The challenge here is to extract important situational updates from these messages, assign them appropriate informational categories, and finally summarize big trove of information in each category. In this paper, we propose a novel framework which first assigns tweets into different situational classes and then summarize those tweets. In the summarization phase, we propose a two stage summarization framework which first extracts a set of important tweets from the whole set of information through an Integer-linear programming (ILP) based optimization technique and then follows a word graph and content word based abstractive summarization technique to produce the final summary. Our method is time and memory efficient and outperforms the baseline in terms of quality, coverage of events, locations *et al.*, effectiveness, and utility in disaster scenarios.

Keywords: Disaster events; Twitter; situational information; classification; summarization.

1. INTRODUCTION

In response to an event, a lot of short messages are posted on social media. Specifically, microblogging platforms such as Twitter provide rapid access to situation-sensitive messages that people post during mass convergence events such as natural disasters. Studies show that these messages contain situational awareness and other useful information such as reports of urgent needs, missing or found people that, if processed timely, can be very effective for humanitarian organizations for their disaster response efforts [27]. Enabling rapid crisis response requires processing of these messages as soon as they arrive. However, typically the volume and velocity of these messages during big disasters can go beyond human processing capacity. For instance, the largest observed peak was during the Sandy hurricane in which around 16 thousands messages per minute were posted using hashtag #Sandy.

Typically, the first step in extracting situational awareness information from these tweets involves classifying them into different informational categories such as infrastructure damage, shelter needs or offers, relief supplies. For instance, one such application is AIDR [10] that performs real-time classification of Twitter messages into different categories. However, even after the automatic classification step, each category still contains thousands of important messages—also increasing each passing minute, which requires further in-depth analysis to make a coherent situational awareness summary for disaster managers to understand the situation.

To get a quick overview of the event and what tweeters are saying about it, a summary of these tweets is very valuable. To deal with the information overload issue and to extract time-sensitive information, in this work, we propose to generate automatic summaries using messages that are classified as useful.

To this end, a straightforward and fast way would be to pick the messages that maximize the coverage of the content words (extractive summarization) [22]. However, to maximize the coverage of information within the specified word limit, it may be necessary to combine related information from several messages (abstractive summarization). For example, consider the following tweets from Nepal earthquake that happened in 2015:

1. `Dharara Tower built in 1832 collapses in Kathmandu during earthquake`
2. `Historic Dharara Tower Collapses in Kathmandu After 7.9 Earthquake`

Both tweets provide information about the collapsing of the Dharara tower. Our objective is to combine important information from both of these tweets and generate a single meaningful situational tweet that contains all the relevant information like, `Dharara tower built in 1832 collapses in Kathmandu after 7.9 earthquake`.

Tweet summarization is a hard problem because given thousands of tweets identifying which tweets are the most important and informative is a subjective problem difficult for even humans to solve. One needs to cover the entire information yet be concise. Even if the most important tweets are chosen, we need to automatically piece them together to create a coherent readable summary. De-

HT '16, July 10-13, 2016, Halifax, NS, Canada

DOI: **http://dx.doi.org/10.1145/2914586.2914600**

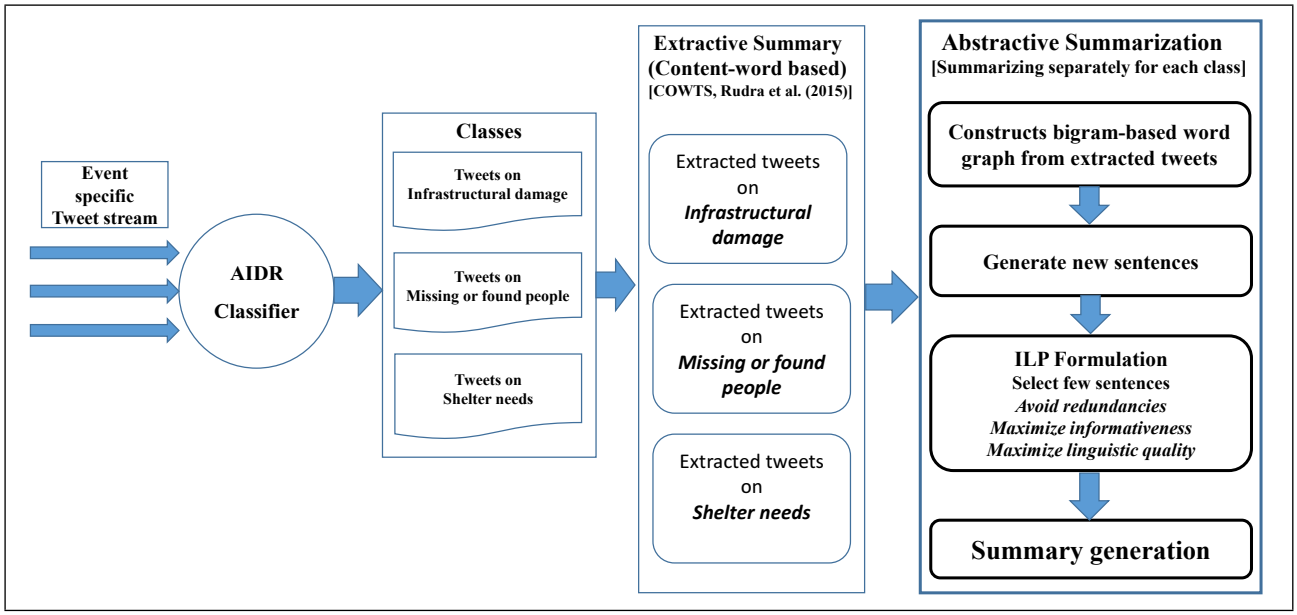


Figure 1: Our proposed framework for Abstractive Summarization of disaster-specific tweets

spite progress in natural language generation, generating abstractive summaries remains a hard problem. Although abstractive summarization [16] produces more compact and informative sentences, the algorithms in general are time-consuming. Hence if the abstractive approach is run over entire incoming set of tweets, it may not be possible to produce the results in run-time (which is one of the important requirements during disaster).

In order to circumvent this problem, first we extract a set of important tweets from the whole set using a fast but effective extractive summarization. In the second step, we use abstractive summarization to choose and rewrite the most important tweets among them, remove redundancy and improve the readability of these tweets.

Figure 1 provides an overview of our proposed approach. We test our proposed approach over Nepal earthquake dataset [14]. In the first step, messages are classified into appropriate classes by AIDR (Artificial Intelligence for Disaster Response) platform [10]. We focus on three information classes, i). *infrastructure and utilities damage*, ii). *missing, trapped, or found people*, and iii). *shelter and supplies needs or offers*. In the summarization phase, first we extract an important set of informative messages from the whole set. Then we propose a word graph based abstractive summarization technique which combines information from semantically similar tweets and finally apply ILP-based[1] content word coverage method to generate final summary for each of the classes respectively.

Our contribution lies in the two-step extractive-abstractive summarization strategy (section 4) that is efficient and yet, generates better summaries with respect to information coverage, diversity, redundancy, coherence, and readability. Experimental results in section 5 confirm that the extractive-abstractive summarization model performs better than state-of-the-art Twitter specific real time summarization models in terms of ROUGE-1 recall and F-scores in most of the cases. We also perform crowdsourcing based experiment and find that our algorithm is superior in terms of readability, information coverage, redundancy and information diversity.

[1]Henceforth we represent integer linear programming approach as ILP-based approach

2. RELATED WORK

Twitter has evolved as one of the most significant sources of information during disaster-specific events. Real-time information posted by affected people on Twitter help improve disaster relief operations [4, 9]. However, it is important to extract the crucial information from the tweets for effective planning by relief organizations [12]. Summarization of Twitter information is significantly more challenging than news articles. The difficulty arises due to two important reasons. First, tweets provide continuous stream of data and therefore it requires real-time processing. Second, the tone of the tweets is different from the formal language used in news articles.

Kedzie, et al. [11] proposed an extractive summarization [6] method to summarize disaster event-specific information from news articles. In contrast, several researchers have attempted to utilize information from Twitter to retrieve important situational updates from millions of posts on disaster-specific events [23, 26, 28, 31]. More recently, sophisticated methods for automatically generating summaries by extracting the most important tweets on the event [15, 22] have been proposed. To generate summaries in real-time, a few approaches for online summarization of tweet streams have recently been proposed [24, 32, 30]. Osborne *et al.* [17] proposed a real event tracking system using greedy summarization. Shou et al. [24] used clustering and LexRank [2] based extractive summarization technique to generate summaries from Twitter.

All the above mentioned methods generate summaries that are merely a collection of tweets. An abstractive summary is desirable because it can generate a summary by collecting important content from the tweets and not including entire tweets. Such a summary should also be more readable than a collection of tweets. Furthermore, the summaries should not contain redundant information. To this end, Olariu [16] proposed a bigram word-graph-based summarization technique that is capable of handling online stream of tweets in real-time and also generate summaries that are abstrac-

tive [18] in nature. Each bigram represents a node in the graph and new words are added real-time from incoming new tweets. However, the method does not consider POS-tag information of nodes and thus can create spurious fusions of tweets having the same bigram but used in a different context. Furthermore, it is a generalized method and does not take into consideration the typicality of disaster related tweets. Banerjee,*et al.* [1] proposed a graph-based abstractive summarization method on news articles. Several new sentences are generated using the graph and an optimization problem is formulated that selects the best sentences from the new sentences to optimize the overall quality of the summary. The optimization problem ensures that redundant information is not conveyed in the final generated summary. However, the graph construction and path generation is computationally expensive in real-time.

In this work, we combine the positive aspects of the above studies - (a) we use a variant of [1] for tweet fusion but employ an extractive step initially to enable the graph to generate new sentences in real-time, (b) we use POS tags along with the words in each bigram to avoid spurious fusions and (c). we also employ disaster-specific content words to determine the importance of a disaster-related tweet [22]. Details of the methodology will be elaborated subsequently.

3. DATASET AND CLASSIFICATION OF MESSAGES

We use the crisis-related messages collected and classified by the AIDR platform [10] from Twitter posted during the 2015 Nepal Earthquake. More than 27 million messages were collected from April 25th to April 27th using different keywords (e.g. "Nepal Earthquake, NepalQuake, NepalQuakeRelief, NepalEarthquake, KathmanduQuake, QuakeNepal, EarthquakeNepal, $\cdots$ "). AIDR is used to classify tweets into several categories (see below); we seek to develop summaries of tweets belonging to each category automatically. For example, for the Nepal earthquake crisis, around 9,000 messages were labeled by the volunteers of the Standby Task Force (SBTF), into the classes/categories specified below. AIDR uses these human-labeled messages to train classifiers that automatically classify subsequent messages collected from Twitter in real-time.

The classes used are as follows:

1. **Injured or dead people:** Casualties due to the crisis
2. **Missing, trapped, or found people:** Questions and/or reports about missing or found people
3. **Displaced people:** People who have been relocated due to the crisis, even for a short time (includes evacuations)
4. **Infrastructure and utilities:** Buildings or roads damaged or operational; utilities/services interrupted or restored
5. **Shelter and supplies:** Needs or donations of shelter and/or supplies such as food, water, clothing, medical supplies or blood
6. **Money:** Money requested, donated or spent
7. **Volunteer or professional services:** Services needed or offered by volunteers or professionals
8. **Animal management:** Pets and animals, living, missing, displaced, or injured/dead
9. **Caution and advice:** Warnings issued or lifted, guidance and tips
10. **Personal updates:** Status updates about individuals or loved ones
11. **Sympathy and emotional support:** Thoughts and prayers
12. **Other relevant information:** Other useful information that helps one understand the situation
13. **Not related or irrelevant:** Unrelated to the situation or irrelevant

In this work, we selected AIDR classified messages from three categories for which the machine confidence was ≥ 0.80. The selected classes and messages in each of the three classes are as follows:

1. **Missing, trapped, or found people** (10,751 machine classified messages)
2. **Infrastructure and utilities** (16,842 machine classified messages)
3. **Shelter and supplies** (19,006 machine classified messages).

4. AUTOMATIC SUMMARIZATION

Given the categorized messages by AIDR for which the machine-confidence score is ≥ 0.80 (as described in section 3), in this section we present our two step automatic summarization approach to generate summaries from each class. We consider the following key characteristics/objectives while developing an automatic summarization approach:

1. A summary should be able to capture most situational updates from the underlying data. That is, the summary should be rich in terms of information coverage.
2. As most of the messages on Twitter contain duplicate information, we aim to produce summaries with less redundancy while keeping important updates of a story.
3. Twitter messages are often noisy, informal, and full of grammatical mistakes. We aim to produce more readable summaries as compared to the raw tweets.
4. The system should be able to generate the summary in real-time, i.e., the system should not be heavily overloaded with computations such that by the time the summary is produced, the utility of that information is marginal.

The first three objectives can be achieved through abstractive summarization and near-duplicate detection, however, it is very difficult to achieve that in real-time (hence violating the fourth constraint). In order to fulfill these objectives, we follow an extractive-abstractive framework to generate summaries. In the first phase (extractive phase), we improve the approach proposed by Rudra et al. [22] and select a sub-set of tweets that cover most of the information produced and then run abstractive summarization over that.

4.1 Extractive Summarization Approach

Disaster-related tweets have distinct features that we use to construct our extractive summaries.

Content Words: As identified in earlier studies [15, 22], in crisis scenarios some specific type of words can play a key role by capturing important events and snapshots. Such useful words which we term as *content words* are as follows:

- Numerals (number of casualties, missing or found people, emergency helpline and ambulance numbers)
- Nouns (capturing important disaster specific context words such as 'hospital', 'ambulance' etc.)
- Information about locations/places surrounding the disaster affected area
- Main verbs ('collapsed', 'destroyed', 'killed', 'trapped' etc.), capturing most of the event phrases

Duplicates: Moreover, a large proportion of messages on Twitter contain redundant information. For instance, in the following five tweets, the same information related to the closure of Kathmandu airport and flights cancellation is conveyed in different ways:

1. `Nepal quake , Kathmandu airport shut, flights from India cancelled via @timesofindia`
2. `Flights to Kathmandu put on hold following powerful earthquake Read more here`
3. `Kathmandu airport shut, flights from India cancelled`
4. `K'mandu airport shut, flights from India cancelled via @timesofindia`
5. `After massive 7.9 earthquake, commercial flights to Kathmandu put on hold`

To handle duplicate or near duplicate information in the messages and to find disaster specific content words we follow two schemes — (i) we remove duplicate and near-duplicate tweets (using the technique developed by Tao, et al., [25]), and (ii) we focus on the content words during summarization as proposed by Rudra, et al. [22].

We consider each class (infrastructure and utilities, missing, trapped or found people, shelter and supplies) separately and try to extract concise summaries for these classes. Specifically, we take day-wise snapshots of each class, i.e., the system produces a summary of the desired length (number of words) over each day for each of the classes using an improved version of COWTS [22]. First we extract a set of content words i.e. words with numeral, noun or verb pos-tags from the messages and try to maximize the coverage of these set of content words. In this phase, our main objective is to capture all the content words within a small number of tweet set such that the next phase of abstractive summarization can generate paths from these tweets and also rank those paths in near real-time. On an average within 1,000 words limit, majority of the content words (present in the entire tweet set) can be covered within the chosen limited set of tweets. We illustrate the rationale behind the 1000 word limit as follows.

Role of *content words* during disaster We want to check whether content words play a different role in disaster scenario or not. We observed that number of content words grow very slowly compared to other general events like sports, music or politics. To understand this, we compare tweet streams posted in above mentioned three different disaster classes (infrastructure, missing, shelter) with those posted during two sports, and specific datewise events;

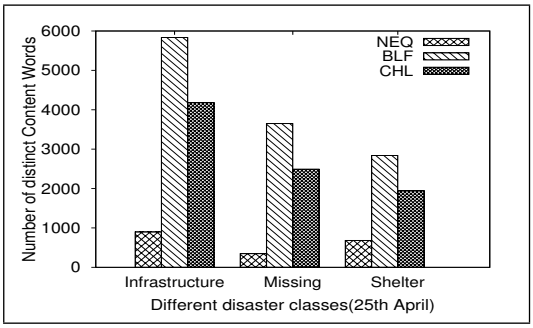


Figure 2: Variation in the number of distinct content words with the number of tweets shown for three disaster classes during Nepal earthquake(NEQ), and two other types of events(BLF, CHL)

these streams were made publicly available by [24]. we have measured the number of content words present in the above mentioned classes. In order to compare their variation with general events, we random sample same number of tweets(as respective disaster classes) from two general events — i) blackfriday (BLF), (ii) chelsea (CHL). Figure 2 shows that number of content words increase very slowly during disaster events compared to any other general event. This observation indicates that capturing such content words can provide an effective coverage of disaster events.

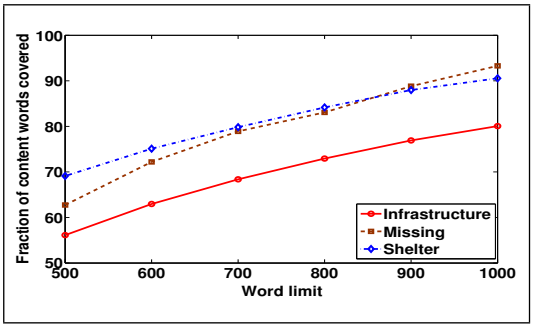


Figure 3: Variation in the coverage of content words with number of extracted tweets.

Content-word coverage vis-a-vis length In Figure 3, we show how the coverage of content words varies with number of tweets extracted from the whole dataset for three different classes of tweets posted on 25th April, 2015. We observe a similar pattern for the other days. An informative set of 1,000 words may be sufficient for the next stage of summarization; hence, we extract a set of tweets with 1,000 word limit constraint in our initial extractive phase of summarization.

After extracting a set of important and informative tweets we try to prepare a more concise and comprehensive summary through a COntent Words based ABstractive Summarization (**COWABS**) approach using these tweets (described next).

4.2 Abstractive Summarization

The goal of this step is to generate an abstractive summary by combining information from multiple tweets. The generated summary must be comprehensive in the sense that it contains more information than extractive summaries of the same length (in words). Our abstractive summarization method is aimed at maximizing the

informativeness of tweets and avoiding redundancy of information jointly. We follow an over-generate and select [29] strategy where we combine multiple tweets to generate a new sentence. Our method tries to select the best sentences from the set of generated sentences and create a summary by optimizing three factors: **Informativeness, Redundancy** and **Readability**. Informativeness and readability have to be maximized, while redundancy is required to be minimized. *Informativeness* is defined as the amount of information in the summary, measured using a centroid-based ranking score. *Redundancy* is minimized such that we do not convey same or similar information in multiple sentences in the summary. We use a trigram-based log-likelihood score using a language model as a dummy representative of the *Readability* of the generated content. We adapt the ILP-based method for summarization proposed by Banerjee, et al. [1] for news summarization; however, we make several modifications to make it usable for tweet summarization. Instead of a unigram-based sentence generation technique, we employ a bigram-based method. This adaptation improves the grammaticality of the resulting summaries. We also introduce a content-word based parameter in the ILP to tackle informativeness and redundancy.

Sentence Generation Process: In order to generate sentences, we **build up a word-graph** [3] with the entire tweet set where each tweet is iteratively added to the graph with the bigrams (adjacent words along with their parts-of-speech (POS) tags[2]) representing the nodes. An edge in the graph represents consecutive words in a sentence. When a new tweet is added to the graph and it contains a bigram that already exists in the graph, the nodes of the new tweet are merged with the existing nodes. We merge the nodes if the words in the bigrams have the same lexical form as well as the same POS tag. POS tags help maintain grammaticality and avoid potentially spurious fusions.

An example of our bigram-based word-graph construction is shown in Figure 4. Each node has been labeled with the form $w1 \shortparallel w2$, where $w1$ and $w2$ refer to the first and the second word in every bigram, respectively. We mark two nodes as the start and the end nodes that indicate the beginning and end of the tweets. The graph is generated considering the following two tweets that were tweeted on a particular day and were assigned to the infrastructure class by the AIDR system — (i) *dharara tower built in 1832 collapses in kathmandu during earthquake*, and (ii) *historic dharara tower collapses in kathmandu after 7.9 earthquake*. We lower-case all words during the graph construction.

Once the graph is formed, sentences, which we term as *tweet-paths* are generated by **traversing paths in the graph** between the dummy *Start* and the *End* nodes. For example, from the graph in Figure 4, we can easily generate a *tweet-path* such as *dharara tower built in 1832 collapses in kathmandu after 7.9 earthquake*. Several such sentences might hold more information than the original tweets, yet containing the same or similar number of words.

We set a minimum (10 words) and maximum (16 words) length for a sentence to be generated. We apply such constraints to avoid very long sentences that might be grammatically ill-formed and very short sentences that are often incomplete. In a real-scenario, the number of generated *tweet-paths* can be several thousands, because there can be multiple points of merging across several tweets. Our goal is to select the best paths from these generated paths with the objective of generating a readable and informative summary. We formulate an ILP problem to select final paths and construct the summary.

[2]We employed a Twitter specific POS tagger [5]. In addition to the regular parts-of-speech tags, it also tags hashtags, retweet mentions, URLs separately. We ignore such words that have these specific hashtags because they are not important in the context of summarization as it might affect readability.

Table 1: Notations used in the summarization technique

Notation	Meaning
L	Desired summary length (number of words)
n	Number of *tweet-paths* considered for summarization (in the time window specified by user)
m	Number of distinct content words included in the n *tweet-paths*
i	index for *tweet-paths*
j	index for content words
x_i	indicator variable for *tweet-paths* i (1 if *tweet-paths* i should be included in summary, 0 otherwise)
y_j	indicator variable for content word j
$Length(i)$	number of words present in *tweet-paths* i
I(i)	Informativeness score of the *tweet-paths* i
LQ(i)	Linguistic quality score of a *tweet-paths*
T_j	set of *tweet-paths* where content word j is present
C_i	set of content words present in *tweet-paths* i

ILP Formulation

The ILP-based technique optimizes based upon three factors - (i) Presence of content words (this is similar to that adopted during the extractive phase): The formulation tries to maximize the number of content words in the final summary. Consequently, maximizing content words automatically implies tackling *redundancy* as constraints avoid choosing the same content words from the set of generated paths. (ii) Informativeness of a path i.e. importance of a path, and (iii) *Linguistic Quality Score* that captures the readability of a path using a trigram confidence score.

Informativeness$(I(i))$**:** We use a centroid based ranking as a proxy of sentence importance as one of the system configurations in our experiments. Centroid-based ranking [19] implies selection of sentences that are more central to the topic of the document. Each sentence is represented as a TF-IDF vector. The centroid is basically the mean of the TF-IDF vectors of all the sentences. Cosine similarity value between the sentences and the centroid is computed and used as the informativeness component in the ILP formulation.

Linguistic Quality Score ($LQ(i)$**) :** The linguistic quality score is computed using a language model. A language model assigns probabilities to the occurrences of words. We use a Trigram language model [8] to compute a score with the goal of assigning higher scores to more probable sequences of words.

$$LQ(s_i) = \frac{1}{(1 - ll(w_1, w_2, w_3, \cdots, w_q))} \quad (1)$$

where $ll(w_1, w_2, w_3, \cdots, w_q)$ is computed as:

$$ll(w_1, w_2, w_3, \cdots, w_q) = \frac{1}{L} log_2 \prod_{t=3}^{q} P(w_t | w_{t-1} w_{t-2}) \quad (2)$$

Assuming the sentence consists of the words $w_1, w_2, w_3, \cdots w_q$, the value of $LQ(i)$ is computed using the above two equations

The summarization of L words is achieved by optimizing the following ILP objective function, whereby the highest scoring *tweet-*

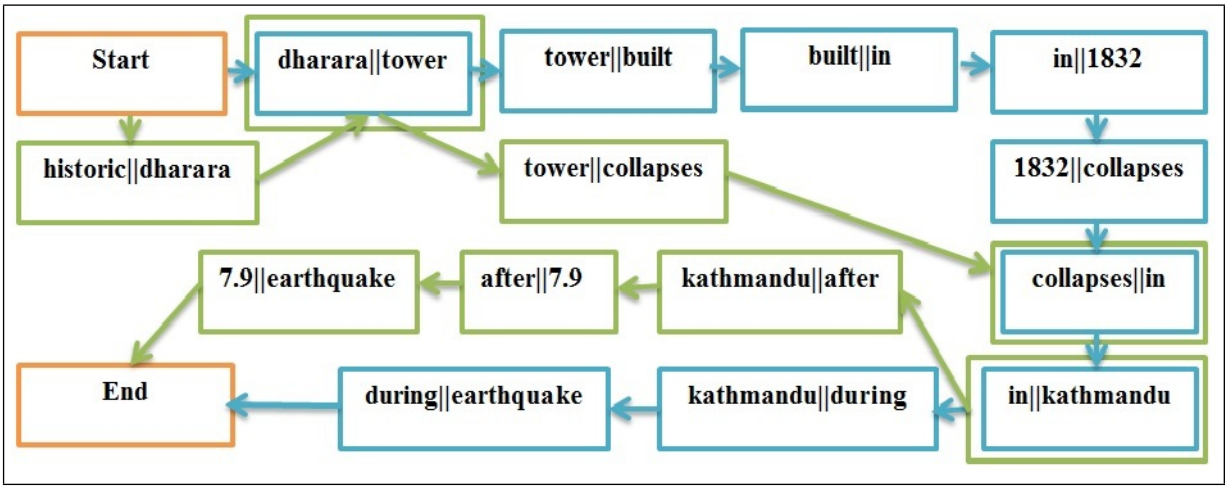


Figure 4: Bigram word graph generated using above two tweets. (We do not show POS tags in the figure for clarity). Nodes from different tweets are represented by different colors. Common nodes contain both the colours. Start and End are special marker nodes.

paths are returned as output of summarization, The equations are as follow:

$$max(\sum_{i=1}^{n} LQ(i).I(i).x_i + \sum_{j=1}^{m} y_j) \quad (3)$$

subject to the constraints

$$\sum_{i=1}^{n} x_i \cdot Length(i) \leq L \quad (4)$$

$$\sum_{i \in T_j} x_i \geq y_j, j = [1 \cdots m] \quad (5)$$

$$\sum_{j \in C_i} y_j \geq |C_i| \times x_i, i = [1 \cdots n] \quad (6)$$

where the symbols are as explained in Table 1. The objective function considers both the number of *tweet-paths* included in the summary (through the x_i variables) as well as the number of important content-words (through the y_j variables) included. The constraint in Eqn. 4 ensures that the total number of words contained in the *tweet-paths* that get included in the summary is at most the desired length L (user-specified) while the constraint in Eqn. 5 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-path* in which this content word is present is selected. Similarly, the constraint in Eqn. 6 ensures that if a particular *tweet-path* is selected to be included in the summary, then the content words in that *tweet-path* are also selected.

We use the GUROBI Optimizer [7] to solve the ILP. After solving this ILP, the set of *tweet-paths* i such that $x_i = 1$, represent the summary at the current time.

5. EXPERIMENTAL SETUP AND RESULTS

In this section, we compare the performance of our proposed framework with state-of-the-art abstractive and disaster-specific summarization techniques. We first describe the baseline technique as well as the experimental settings.

5.1 Experimental Settings

Given the AIDR classified messages from three classes i.e. (1) *infrastructure and utilities damage*, (2) *missing, trapped, or found people*, and (3) *shelter and supplies*, we perform date-wise split starting from 25th April to 27th April, 2015 of the messages.

Establishing gold standard summaries: We take summaries generated by experts from the disaster management domain. During Nepal earthquake, UN OCHA (United Nations Office for the Coordination of Humanitarian Affairs) among other humanitarian organizations used AIDR's output (i.e. machine classified messages) for their disaster response efforts. In this case, the experts were given the machine classified messages that they analyzed to generate a situational awareness report for each informational category. We consider these reports as our gold standard summaries, which contain 498, 4,609, and 6,826 words for infrastructure, missing, and shelter classes respectively. Following their standard practice, the experts also incorporated useful information from other social media sources such as Facebook in the reports.

Baseline approaches: We use three state-of-the-art summarization approaches as our baseline that are described below:

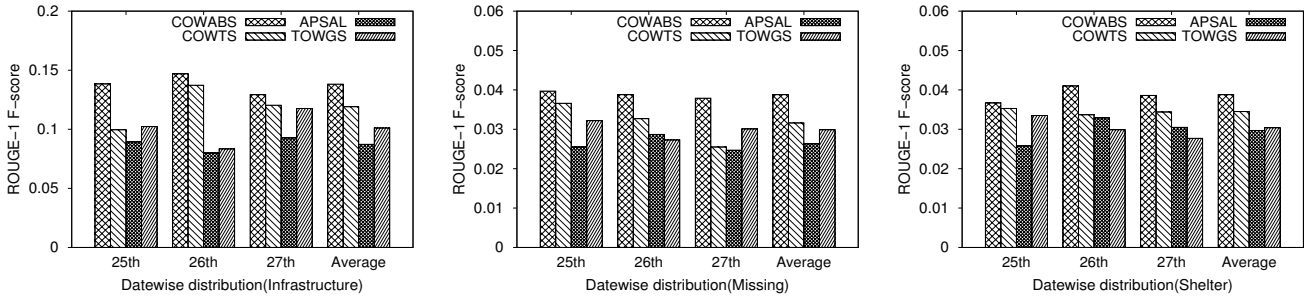
1. **COWTS:** is an extractive summarization approach specifically designed for generating summaries from disaster-related tweets [22].

2. **APSAL:** is an affinity clustering based summarization technique proposed by Kedzie et al. [11]. It mainly considers news articles and focuses on human-generated information nuggets to assign salience score to those news articles while generating summaries.

3. **TOWGS:** is an online abstractive summarization approach proposed by Olariu [16]. It is designed for informal texts like tweets. They consider bigrams as nodes and build word graph using these nodes. To generate a summary, they start from most frequent bigrams to explore different paths. However, TOWGS method is not proposed to generate event-specific summaries. In our case, we modified it to generate event-specific summaries. While generating a path, we start with most frequent bigram node, and subsequently expand the path by finding most promising adjacent node based on co-occurrence frequency etc. as proposed by Olariu. We prepare a final summary by coalescing the generating paths,

Table 2: Comparison of ROUGE-1 recall (with classification, Twitter specific tags, emoticons, hashtags, mentions, urls, removed and standard rouge stemming(-m) and stopwords(-s) option) for COWABS (the proposed methodology) and three baseline methods (COWTS, APSAL, TOWGS) on the same situational tweet stream across three different classes (infrastructure, missing, shelter) over three different dates.

Step size	ROUGE-1 recall Score											
	Infrastructure				Missing				Shelter			
	COWABS	COWTS	APSAL	TOWGS	COWABS	COWTS	APSAL	TOWGS	COWABS	COWTS	APSAL	TOWGS
25th	**.0972**	.0678	.0588	.0656	**.0206**	.0189	.0131	.0156	**.0189**	.0185	.0131	.0181
26th	**.1018**	.0927	.0520	.0588	**.0201**	.0168	.0147	.0140	**.0211**	.0173	.0168	.0152
27th	**.0882**	.0791	.0610	.0701	**.0196**	.0131	.0126	.0138	**.0198**	.0176	.0155	.0141

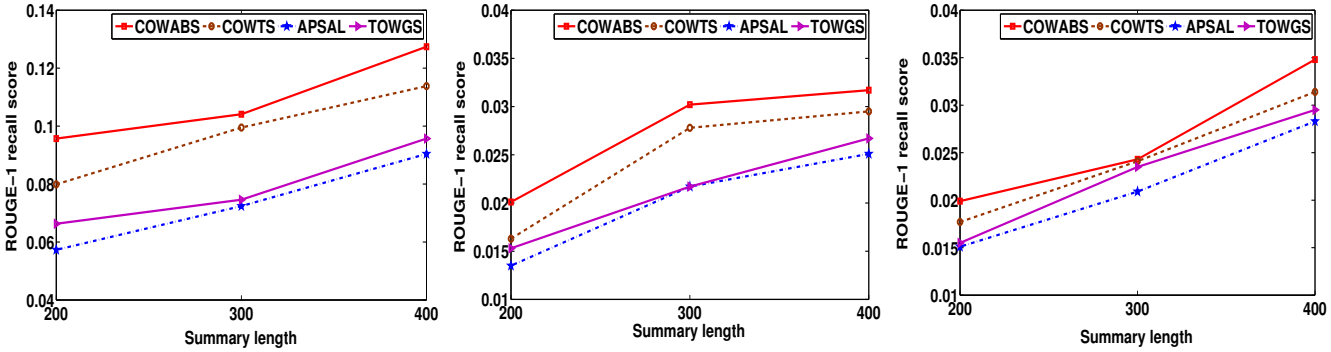


(a) Infrastructure **(b) Missing** **(c) Shelter**

Figure 5: ROUGE-1 F-scores of the date wise summaries of different classes, generated by the proposed methodology (COWABS) and three baseline methods from 25th to 27th April, 2015.

Table 3: Summary of length 50 words(excluding #,@,RT,URLs), generated from the situational tweets of the infrastructure class (26th April) by (i) COWABS (proposed methodology), (ii) COWTS.

Summary by COWABS	Summary by COWTS
Times of india live blog earthquake in katmandu , 25 04 2015. Chairs follow-up meeting to review situation following earthquake in decades. 5 commercial flights have landed in kathmandu was painted in 1850 ad. Iaf's c-130j aircraft carrying 55 passengers , including four infants , lands at delhi's palam airport. Nepal quake photos show historic buildings reduced to rubble as survivor search continues.	#PM chairs follow-up meeting to review situation following #earthquake in #Nepal @PMOIndia #nepalquake. @SushmaSwaraj @MEAcontrolroom Plz open help desk at kathmandu airport. @Suvasit thanks for airport update. #NepalQuake. Pakistan Army Rescue Team comprising doctors, engineers & rescue workers shortly after arrival at #Kathmandu Airport http://t.co/6Cf8bgeort. RT @cnnbrk: Nepal quake photos show historic buildings reduced to rubble as survivor search continues. http://t.co/idVakR2QOT http://t.co/Z.



(a) Infrastructure **(b) Missing** **(c) Shelter**

Figure 6: Variation in ROUGE-1 recall scores with system summary length

however, similar tweet-paths are removed based on cosine similarity.

Evaluations: We perform two types of evaluations. First, we use the standard ROUGE [13] metric for evaluating the quality of summaries generated using the proposed as well as the baselines 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 paid crowdsourcing (described below).

5.2 Performance comparison

Table 2 and Figure 5 depict the ROUGE-1 recall and F-scores for the three algorithms for each class and day. *Note that the low recall values correspond to the mismatch in the length of the system generated summary (200 words) and gold standard summary (ranging from 500 to 7000 words)*. We can see that COWABS performs significantly better compared to other three baselines - the improvement ranges from 20% to 40%.

Further in order to test the robustness of COWABS, we compare the performance of the baselines by increasing the summary length. To perform this experiment, we vary summary length in the range of 200 to 400 for all the different classes. From figure 6, it is observed that as summary length increases, COWABS is increasingly able to capture more informative content compared to other baseline approaches.

To give a flavor of the kind of summaries produced by the proposed summarization approach, Table 3 shows summaries generated by COWABS and COWTS (both disaster-specific methodologies) from the same set of messages (i.e tweets form infrastructure class posted on 26th April). The two summaries are quite distinct. We find that summary returned by COWABS is more informative and diverse in nature compared to COWTS. For instance, we can see the COWABS summary contains information about flights, damages of buildings, and information sources.

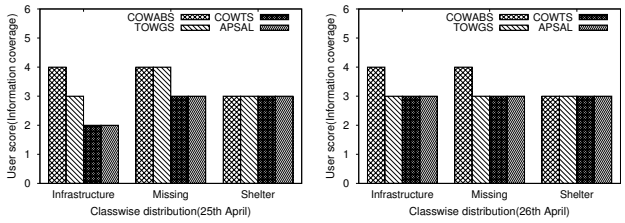
Redundancy in summaries: Apart from ROUGE-1 score, we also measure redundancy score of the summaries as this can indicate if the summaries contain distinct or redundant information. We compute redundancy score for a summary as follows: For each sentence included in the summary (a sentence can be a tweet or a path), we assign it a *sentence redundancy score* as its maximum cosine similarity (excluding #,@,URLs,stopwords) with any other sentence in the summary. Finally, we take an average of the individual sentence redundancy scores to compute the redundancy value for the summary. Table 4 shows redundancy values of different methods across each of the three classes. Through abstraction, we can reduce redundancy by 30.11%, 27.75%, 47.95% respectively for each of the three classes.

Table 4: Redundancy score for different methods of summarization (lower is better)

Method	Redundancy score		
	Infrastructure	Missing	Shelter
COWABS	0.1775	0.2099	0.1433
COWTS	0.1833	0.2122	0.2112
APSAL	0.2986	0.3797	0.3731
TOWGS	0.3205	0.3222	0.3336

Evaluation using crowdsourcing: Next, we perform crowdsourced evaluation using the CrowdFlower [3] crowdsourcing platform. We take summaries generated from each class using our proposed method and all three baselines for each day—in total we use 9 summaries. A crowdsourcing task, in this case, consists of four summaries (i.e. one proposed and three from baseline methods) and the four criteria with their description (as described below) along with a scale from 1 (very bad) to 5 (very good) for each criterion. For each task, we asked five different annotators to read each summary carefully and provide scores for each criterion. The exact description of the crowdsourcing task is as follows: *"The purpose of this task is to evaluate machine-generate summaries using tweets collected during the Nepal Earthquake happened in 2015. Each task given below has 4 summaries of length 200 words generated by 4 different algorithms on same set of tweets (thousands in this case) belong to a particular topic. Given the summaries and their topic, we are interested in comparing them based on the following criteria: Information coverage, Redundancy, Diversity and Readability"*. We provide details analysis of our crowdsourcing task in the following section:

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. Our proposed method is able to capture very good situational information/updates in case of Infrastructure and Missing class for both of the days chosen while it performs fairly in the shelter class. However, in 5 cases it performs better than the three competing techniques and for rest of the 4 cases it performs equally well to some baseline technique. Figure 7 shows details ranking of users for 25th and 26th April [4].



(a) 25th (info. coverage) **(b)** 26th (info. coverage)

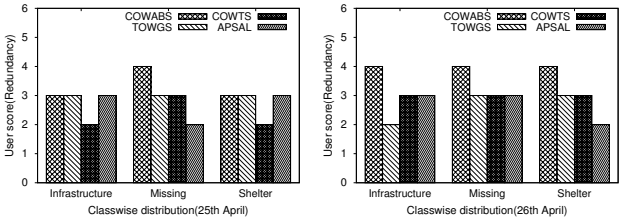
Figure 7: Results of the crowdsourcing based evaluation based on the information coverage

Redundancy corresponds to the duplication of same information. A good summary should be representative of underlying data and should have less redundant information. COWABS outperforms other baselines in case of 6 summaries and in rest of the cases it performs equally well to some other baseline. In our first phase of extractive summarization technique, we try to remove similar tweets to reduce redundancy and user observations suggest that COWABS is taking advantage of that phase to reduce redundancy in final summary. We provide details of user ranking in figure 8.

Diversity corresponds to the novelty of sentences in a summary. A good summary should contain diverse informative sentences. Although we do not apply any direct parameter in our ILP framework to control diversity, but in our abstractive ILP method, we not only rely on importance score of paths but also coverage of different content words which helps in capturing information from various

[3] `http://www.crowdflower.com/`

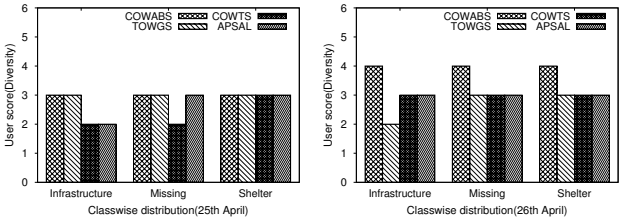
[4] We only keep two dates to maintain clarity and brevity



(a) 25th (redundancy) **(b)** 26th (redundancy)

Figure 8: Results of the crowdsourcing based evaluation based on the redundancy

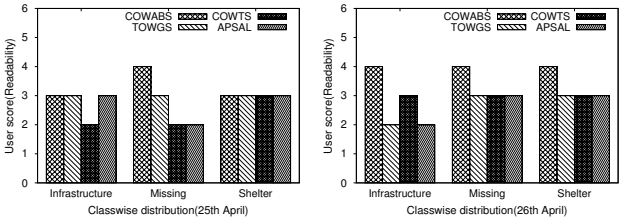
dimensions. This is also quite clear from figure 9. Out of 9 summaries, COWABS appeared to be more diverse in 4 cases and in all other cases it performs equally well as others.



(a) 25th (info. diversity) **(b)** 26th (info. diversity)

Figure 9: Results of the crowdsourcing based evaluation based on the information diversity

Readability corresponds to the overall readability of the content in a summary. For instance, a good summary should be easily readable, well formed, having less grammatical mistakes. One of our main focus in this summarization technique is to make summaries more readable and coherent. For that, we have applied linguistic quality score in our final ILP framework and prefer those paths which have higher linguistic scores. According to user evaluations, COWABS appears to be more readable compared to other baselines in 6 cases. Figure 10 reveals that readability wise our summaries get lowest score as 3, the performance is particularly good on 26th April where it is marked 4 (good) for all the cases.



(a) 25th (readability) **(b)** 26th (readability)

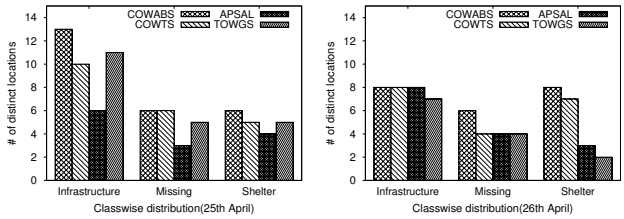
Figure 10: Results of the crowdsourcing based evaluation based on the readability

In particular, COWABS outperforms other baseline techniques in most of the cases and rest of the cases it is a tie but it never performs poorly compared to any baseline method.

Evaluation in terms of time taken: During such crisis scenario, time is very critical and one of our main focuses is on real-time summarization. Hence, we analyze the execution times of the various techniques. Table 5 provide details information about run-time of our proposed COWABS method and other three baselines. The time taken is comparable with other real-time summarization approaches like COWTS [22] and TOWGS [16]. APSAL requires more time due to non-negative matrix factorization and affinity clustering approach over large dataset. COWABS is taking slightly higher running time compared to COWTS and TOWGS, but it is at par with these two baselines. However, COWABS performs much better compared to COWTS and TOWGS in terms of information coverage, readability, redundancy, diversity.

Quality of Information Summarized: Beyond the mere numbers proving our superiority, we also looked into the tweets and checked its quality with respect to (a). number of distinct places mentioned (b). number of event phrases used and (c). extent of numbers present in the summary. Details of which follow -

Location coverage: During large scale disaster like earthquake, flood *et al.* several parts of a country are damaged and coverage of information from these different places are necessary. Location coverage corresponds to the information about different places a summary contains. For instance, a summary with diverse information from many locations is considered better in terms of location coverage. The problem is challenging in the sense that there is overwhelmingly more information about big cities/towns in the tweet. For example, during Nepal earthquake most of the information are available in Twitter from its capital city **Kathmandu** but there is a scarcity of information from local villages like **Barpak**, **Lamjung** etc. Our proposed method COWABS is able to capture information about more number of locations in 7 cases. Figure 11 shows number of locations captured by different methods.



(a) 25th (location coverage) **(b)** 26th (location coverage)

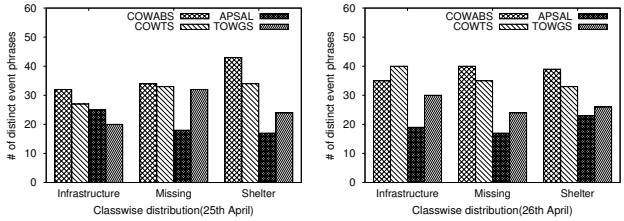
Figure 11: Coverage of locations in summaries by COWABS (proposed) and three baselines (COWTS, APSAL, TOWGS)

Event coverage: To extract event phrases, we have used a named entity recognition tool designed explicitly for tweets [20, 21], which tags the words in a tweet with event. For instance, in the tweet "180 Bodies Retrieved From Debris of Nepal's Historic Tower", the word 'Retrieved' is tagged as an event phrase. A good summary should contain more number of distinct events. Out of the 9 cases, our proposed method is able to capture more number of events in 7 cases. We provide details information about event coverage in figure 12.

Numerical coverage: As identified in earlier studies [22, 15] numerals play a key role in disaster scenario as they contain information about casualties, injured or missing people, tracking numbers, helpline information etc. Summary which contains more numer-

Table 5: Runtime (seconds) of different algorithms for each of the three classes (infrastructure, missing, shelter).

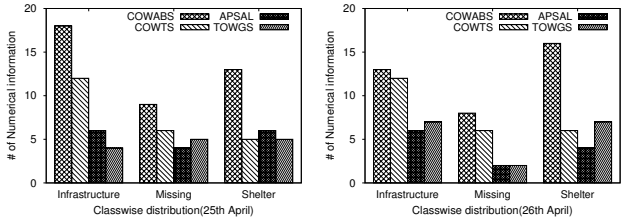
Date	Infrastructure					Missing					Shelter				
	#Tweets	COWABS	COWTS	APSAL	TOWGS	#Tweets	COWABS	COWTS	APSAL	TOWGS	#Tweets	COWABS	COWTS	APSAL	TOWGS
25/04	9371	25.57	23.46	1187.19	19.34	3953	14.98	11.15	35.20	7.10	2593	11.21	8.68	96.95	7.22
26/04	5036	19.14	17.21	4507.50	18.91	5668	16.98	14.89	504.41	14.24	11178	42.45	40.24	16002.47	29.84
27/04	2435	11.07	8.76	533.62	14.17	1130	7.90	5.54	21.70	4.90	5267	23.94	21.21	7653.34	22.73



(a) 25th (event coverage) **(b)** 26th (event coverage)

Figure 12: Coverage of event phrases in summaries by COWABS (proposed) and three baselines (COWTS, APSAL, TOWGS)

ical information is more useful during disaster scenario, specially for certain types of disaster classes like 'missing or trapped people', 'injured or dead people' etc. Our proposed method is able to capture more number of numerical information in 8 cases. Figure 13 provides details about numerical information coverage of different summarization techniques.



(a) 25th (numeral coverage) **(b)** 26th (numeral coverage)

Figure 13: Coverage of numerals in summaries by COWABS (proposed) and three baselines (COWTS, APSAL, TOWGS)

It is quite clear that COWABS is able to capture more microlevel information compared to other baseline techniques which is a crucial requirement of summarization process.

5.3 Reason behind better performance

We try to dissect the three baseline algorithms and identify their limitations and thus understand the reason behind COWABS superior performance.

TOWGS, proposed by Olariu [16], ranks different paths based on normal term frequency of each node (bigram) constituting a path. However, term weight might not clearly justify relevancy of a term. Besides this, Olariu's technique does not employ any grammaticality or linguistic quality check; as a result, TOWGS suffers due to poor readability as evident from our crowdsourcing experiments. Our other baseline, APSAL, is a semi-supervised technique which also maintains clusters of related information and finally chooses one exemplar tweet from each cluster. All clusters are assumed to have equal importance which might not always be applicable. Furthermore, the method was originally proposed for summarization on formal news articles where clusters are more coherent as compared to clusters of tweets. Due to mismatch in reality, it is not able to generate summaries with high informative content (as evident from ROUGE-1 scores). Furthermore, as can be seen from the human judgments, APSAL summaries also contain significant redundancies. COWTS although extractive performs the best among all the baselines according to the ROUGE-1 scores perhaps due to its simplicity. However, COWTS suffers from the fundamental problem of extractive summarization namely redundancy. Same or similar information might exist in two different tweets, yet they can be the part of the summary. As a result, information is repeated as evident from our crowdsourcing experiments.

6. CONCLUSION

A large number of tweets are posted during disaster scenarios and a concise, categorical representation of those tweets is necessary. In this paper, we develop a complete system to generate summaries in real time from the incoming stream of tweets. We specifically take the tweets generated during Nepal Earthquake and generate comprehensive abstractive summaries for three most important classes - infrastructure, missing and shelter. We perform an extensive evaluation of our algorithm by roping in disaster-related experts in the loop - results show that our method - COWABS perform significantly better than all competing baselines. We also perform crowd-sourcing experiment asking the crowd to rank our algorithm compared to baselines – in all the cases ours is ranked higher. Also independently the crowd comments that our summaries are high-quality thus satisfying the purpose for which the entire exercise has been undertaken. We would strive to deploy the system so that it can be practically used for any future disaster event.

Acknowledgement: This research was partially supported by a grant from the Information Technology Research Academy (ITRA), DeITY, Government of India (Ref. No.: ITRA/15 (58)/ Mobile/ DISARM/ 05) Additionally, K. Rudra was supported by a fellowship from Tata Consultancy Services. The authors thank the anonymous reviewers whose suggestions greatly helped to improve the paper.

7. REFERENCES

[1] S. Banerjee, P. Mitra, and K. Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[2] G. Erkan and D. R. Radev. LexRank:Graph-based lexical centrality as salience in text summarization. *Artificial Intelligence Research*, 22:457–479, 2004.

[3] K. Filippova. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics, 2010.

[4] H. Gao, G. Barbier, and R. Goolsby. Harnessing the crowdsourcing power of social media for disaster relief. *Intelligent Systems, IEEE*, 26(3):10–14, 2011.

[5] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. Smith, A. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. ACL*, 2011.

[6] V. Gupta and G. S. Lehal. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3):258–268, 2010.

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

[8] K. Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics, 2011.

[9] M. Imran, C. Castillo, F. Diaz, and S. Vieweg. Processing social media messages in mass emergency: a survey. *ACM Computing Surveys (CSUR)*, 47(4):67, 2015.

[10] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg. AIDR: Artificial intelligence for disaster response. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 159–162. International World Wide Web Conferences Steering Committee, 2014.

[11] C. Kedzie, K. McKeown, and F. Diaz. Predicting salient updates for disaster summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1608–1617, Beijing, China, July 2015. Association for Computational Linguistics.

[12] B. Klein, X. Laiseca, D. Casado-Mansilla, D. López-de Ipiña, and A. P. Nespral. Detection and extracting of emergency knowledge from twitter streams. In *Ubiquitous Computing and Ambient Intelligence*, pages 462–469. Springer, 2012.

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

[14] 2015 Nepal earthquake – Wikipedia, April 2015. http://en.wikipedia.org/wiki/2015_Nepal_earthquake.

[15] M.-T. Nguyen, A. Kitamoto, and T.-T. Nguyen. TSum4act: A Framework for Retrieving and Summarizing Actionable Tweets during a Disaster for Reaction. In *Proc. PAKDD*, 2015.

[16] A. Olariu. Efficient online summarization of microblogging streams. In *Proc. EACL*, pages 236–240, 2014.

[17] M. Osborne, S. Moran, R. McCreadie, A. V. Lunen, M. Sykora, E. Cano, N. Ireson, C. Macdonald, I. Ounis, Y. He, T. Jackson, F. Ciravegna, and A. OBrien. Real-Time Detection, Tracking, and Monitoring of Automatically Discovered Events in Social Media. In *Proc. ACL*, 2014.

[18] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[19] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[20] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *EMNLP*, 2011.

[21] A. Ritter, Mausam, O. Etzioni, and S. Clark. Open domain event extraction from twitter. In *KDD*, 2012.

[22] K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh. Extracting situational information from microblogs during disaster events: a classification-summarization approach. In *Proc. CIKM*, 2015.

[23] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proc. World Wide Web Conference (WWW)*, pages 851–860, 2010.

[24] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: Continuous summarization of evolving tweet streams. In *Proc. ACM SIGIR*, pages 533–542, 2013.

[25] K. Tao, F. Abel, C. Hauff, G.-J. Houben, and U. Gadiraju. Groundhog Day: Near-duplicate Detection on Twitter. In *Proc. World Wide Web Conference (WWW)*, pages 1273–1284, 2013.

[26] S. Verma, S. Vieweg, W. J. Corvey, L. Palen, J. H. Martin, M. Palmer, A. Schram, and K. M. Anderson. Natural Language Processing to the Rescue? Extracting "Situational Awareness" Tweets During Mass Emergency. In *Proc. AAAI ICWSM*, 2011.

[27] S. Vieweg, C. Castillo, and M. Imran. Integrating social media communications into the rapid assessment of sudden onset disasters. In *Social Informatics*, pages 444–461. Springer, 2014.

[28] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness. In *Proc. ACM SIGCHI*, 2010.

[29] M. A. Walker, O. Rambow, and M. Rogati. Spot: A trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8. Association for Computational Linguistics, 2001.

[30] Z. Wang, L. Shou, K. Chen, G. Chen, and S. Mehrotra. On summarization and timeline generation for evolutionary tweet streams. *IEEE Transactions on Knowledge and Data Engineering*, 27:1301–1314, 2015.

[31] W. Xu, R. Grishman, A. Meyers, and A. Ritter. A preliminary study of tweet summarization using information extraction. *NAACL 2013*, page 20, 2013.

[32] A. Zubiaga, D. Spina, E. Amigo, and J. Gonzalo. Towards Real-Time Summarization of Scheduled Events from Twitter Streams. In *Hypertext(Poster)*, 2012.