Summarizing Situational Tweets in Crisis Scenarios: An Extractive-Abstractive Approach

Koustav Rudra, Pawan Goyal, Niloy Ganguly, Muhammad Imran, and Prasenjit Mitra

Abstract—Microblogging platforms such as Twitter are widely used by eyewitnesses and affected people to post situational updates during mass convergence events such as natural and man-made disasters. These crisis-related messages disperse among multiple classes/categories such as infrastructure damage, shelter needs, information about missing, injured, and dead people etc. Side by side, we observe that sometimes people post information about their missing relatives, friends with details like name, last location etc. Such kind of information is time-critical in nature and their pace and quantity do not match with other kind of generic situational updates. Also, requirement of different stakeholders (government, NGOs, rescue workers etc.) vary a lot. This brings two-fold challenges — (i). extracting important high-level situational updates from these messages, assign them appropriate categories, finally summarize big trove of information in each category and (ii). extracting small-scale time-critical sparse updates related to missing or trapped persons. In this paper, we propose a classification-summarization framework which first assigns tweets into different situational classes and then summarizes those tweets. In the summarization phase, we propose a two stage *extractive-abstractive* summarization framework. In the first step, it extracts a set of important tweets from the whole set of information, develops a bigram-based word-graph from those tweets, and generates paths by traversing the word-graph. Next, it uses an Integer-linear programming (ILP) based optimization technique to select the most important tweets and paths based on different optimization parameters such as informativeness, coverage of content words etc. Apart from general class-wise summarization, we also show the customization of our summarization model to address time-critical sparse information needs (e.g., missing relatives). Our proposed method is time and memory efficient and shows better performance than state-of-the-art methods both in terms of quantitative and qualitative judgement.

Index Terms—Crisis; Microblogs; Twitter; humanitarian classes; classification; summarization; missing persons; content words

1 Introduction

The widespread adoption of mobile and communication technologies is increasing traffic on social media platforms such as Twitter and Facebook, in particular during natural and man-made disasters large volume of situational messages are shared on Twitter by eyewitnesses and bystanders. Recent studies [1], [2], [3] showed that these situation-sensitive messages contain diverse and important information including reports of 'infrastructure damage', 'affected, stranded, or trapped people', 'urgent needs of victims' among others. Apart from situation-related updates, many uninformative and irrelevant messages are also posted, which contain personal opinion, sentiment of people [1], and advertisements. Timely processing of disaster-related messages on social media can be very effective for humanitarian organizations (United Nations' OCHA, RedCross etc.) for their disaster response efforts [4]. However, enabling rapid crisis response requires processing of these messages as soon as they arrive, which is difficult for humans to manually process as large volume of information is posted at a rapid pace during disaster. Hence, it is necessary to develop automated methods to extract, analyze, and summarize situational information during disasters in *real-time*. Typically, the first step in extracting situational awareness information from these tweets involves classifying them into different humanitarian classes such as infrastructure damage, shelter needs or offers, relief supplies etc. For instance, one such application is AIDR [5] that performs real-time classification of Twitter messages into different informational classes. However, even after the automatic classification step, each class still contains thousands of 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. Several recent studies [1], [3], [6] tried to summarize the information posted during crisis. However, all of these methods tried to select informative tweets based on some criteria to represent summary at a particular instant (*extractive summarization*). For example, Rudra et al. [1] proposed a simple and effective extractive summarization method which tries to capture informative content words in the summary. However, during disaster, lots of tweets

- *K. Rudra is with the L3S Research Center, Leibniz University Hannover, 30167 Hannover, Germany (e-mail: rudra@l3s.de). P. Goyal and N. Ganguly are with the Department of Computer Science and Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: pawang@cse.iitkgp.ac.in; niloy@cse.iitkgp.ac.in). M. Imran is with the Department of Social Computing, Qatar Computing Research Institute, Hamad Bin Khalifa University (HBKU), Doha, Qatar (e-mail: mimran@hbku.edu.qa). P. Mitra is with the College of Information Science and Technology, Pennsylvania State Univer- sity, State College, PA 16801 USA (e-mail: pmitra@ist.psu.edu).*

- *This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under Grant 832921. (Corresponding author: Koustav Rudra.)*
- *DOI: 10.1109/TCSS.2019.2937899*
- *Link to the published version: https://ieeexplore.ieee.org/document/8839735*

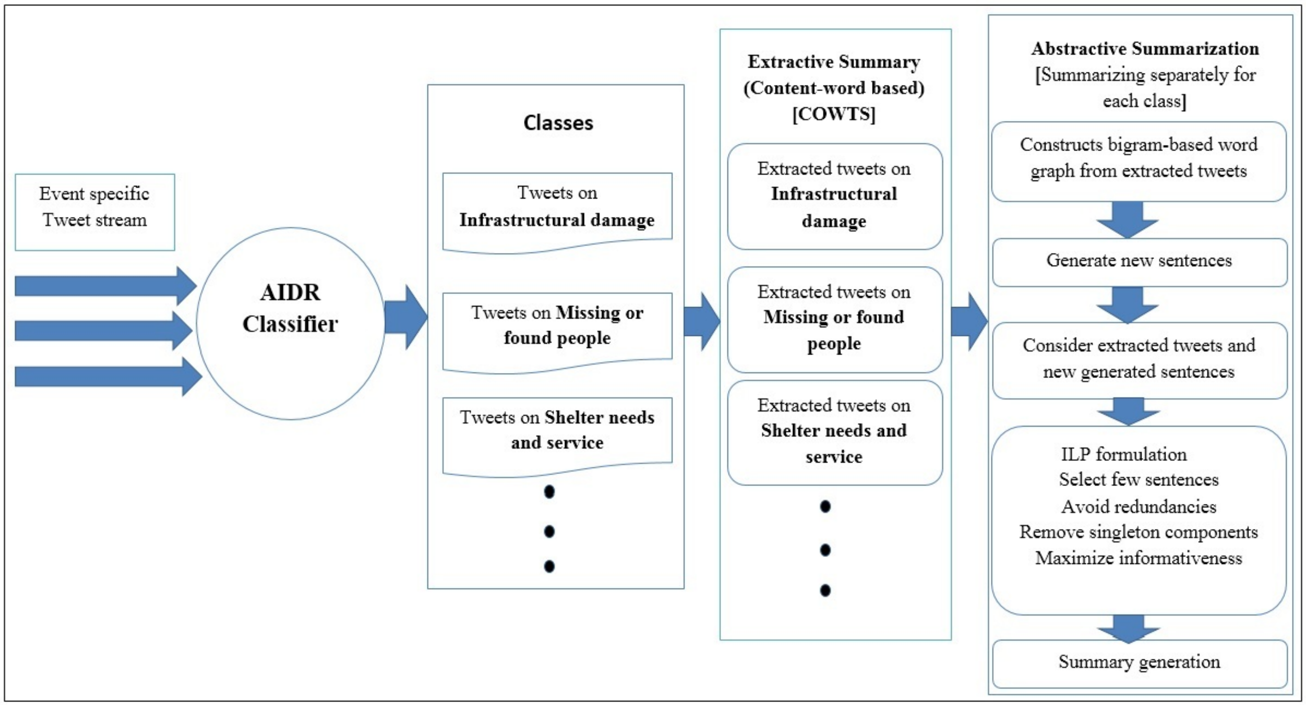

Event specific
Tweet stream
AIDR
Classifier
Classes
Tweets on Infrastructural damage
Tweets on Missing or found people
Tweets on Shelter needs and service
Extractive Summary (Content-word based) [COWTS]
Extracted tweets on Infrastructural damage
Extracted tweets on Missing or found people
Extracted tweets on Shelter needs and service
Abstractive Summarization [Summarizing separately for each class]
Constructs bigram-based word graph from extracted tweets
Generate new sentences
Consider extracted tweets and new generated sentences
ILP formulation
Select few sentences
Avoid redundancies
Remove singleton components
Maximize informativeness
Summary generation

Fig. 1: Our proposed Extractive-Abstractive summarization framework for disaster-specific tweets.

are posted which are duplicates or near duplicates of each other [2] and combining information from multiple related tweets helps to cover more information within a specified word limit (abstractive summarization). For example, consider the following tweets from the Nepal earthquake event that happened in 2015:

1) `Tribhuvan international airport is closed.`
2) `Airport is closed due to 7.9 earthquake.`

We get information about the closure of the airport from both the tweets. Our objective is to combine important information from both of these tweets and generate a single meaningful situational message that contains all the relevant information like, ``Tribhuvan international airport is closed due to 7.9 earthquake''.

Summarization of evolving tweet stream is in general a hard problem because selecting an important subset of tweets in real-time is a subjective task and it is difficult even for humans. The objective is to select important tweets, gather pieces of information, combine them, and prepare a concise report. Despite progress in natural language generation, researchers still face problem in generating abstractive summaries. Also, abstractive algorithms are time consuming in nature; hence, it may be difficult to generate summaries in real-time from large volume of tweet stream (which is one of the important requirements during disaster).

In order to circumvent this problem, in this paper, first we extract a set of important tweets from the given set of tweets using an effective and fast extractive summarization technique. In the second step, we use abstractive summarization to select and combine information from multiple related tweets, so as to remove redundancy.

In addition to the general situational awareness, some stakeholders, crisis responders, and rescue workers may also want to get updates at a much finer granularity with a specific focus on events, persons, and locations connected with the disaster. For example, one may not only be interested in 'missing people', but, more specifically, they may be interested in information about the Australian mountain climbers who were at the foothills of Mt. Everest when the earthquake hit Nepal. We observe that people post information like 'name', 'last seen location' etc. about their missing relatives during crisis and ask help from rescue workers. A general summarization framework may not be able to capture such small-scale information needs. Hence, in this paper, we propose a separate summarization framework to retrieve relevant information about such missing victims. The objective of this work is two fold — (i). classify situational tweets into different humanitarian classes and generate concise summary for each of these classes to ease the task of emergency responders, and (ii). handle sparse information needs (e.g., missing relatives).
Our major contributions are listed as follows:

- We propose a novel extractive-abstractive summarization framework which satisfies two major requirements (i.e., information coverage, real-time) during disasters. Specifically, we perform the following steps to generate a summary: (i) Tweet stream is automatically classified into various humanitarian classes using AIDR [5] with an objective to produce a coherent summary for each humanitarian class. However, due to the real-time constraint, applying abstractive summarization method over the entire tweets of a specific class is not efficient. Hence, we first apply a disaster-specific *extractive* summarization approach COWTS [1] to extract a concise and important set of informative messages from the whole set. (ii) Next, we develop a word-graph using the tweets extracted in the first step. In the word graph, we consider bigrams (consecutive words with pos-tag information) as nodes to handle noisy nature of tweets. After that, we generate paths by traversing this graph (abstractive phase). (iii) Finally, we consider tweets and paths for each of the classes and apply an Integer Linear Programming(ILP) based[1] summarization technique which tries to maximize the coverage of content words (nouns, numerals, and verbs) in the final summary. In the second step, we consider bigrams with POS-tag information as nodes to prevent generation of spurious paths. However, this step also reduces the probability of fusion of nodes which in turn results in the loss of information. Hence, in this paper, we consider *tweets* along with *paths* to avoid information loss.

- Tweets are quite informal in nature and contain more than one component (in sentences) [1], [7]. For example, tweet "Breaking: 7 people died in the blast" contains two different components, i.e., "breaking" and "7 people died in the blast". Here, "breaking" is a singleton component. However, we observe that such singleton components do not contain effective information regarding the disaster and may be removed from the list of *content words*.
- We observe that in many cases, general summarization models fail to cover time-critical sparse information such as personal information about missing relatives. In our last contribution, we show a way to customize our general proposed model to generate such specific summaries and propose a named-entity-recognizer [8] based summarization method to extract and summarize such information (Section 4.3).

Note that, our summarization approach was first proposed in a prior study [2]. The present work improves our prior work as follows. First, we improve the methodology of COWABS in [2] and show that the new methodology (i.e., COWEXABS (Section 4)) proposed in the present work outperforms COWABS. Earlier, in COWABS, we only rely on paths to generate the final summary. However, we observe that paths are generated if tweets have common bigram. Relying only on paths may result in loss of information. Hence, in this paper, we consider both raw tweets and paths in the final summarization stage. Second, in COWABS, we did not consider the importance (weight) of content words. In this paper, we consider importance of content words in the ILP framework (Section 4). Third, we remove singleton components from the list of *content words* and experimental results (Section 5) suggest that removal of such noise helps improve the quality of the final summary. Our contribution lies in the two-step extractive-abstractive summarization strategy (Section 4) that is efficient and generates better summaries. Finally, we propose a named-entity tagger based summarizer to collect small scale information about missing persons. To the best of our knowledge, this is the first attempt to extract such kind of small scale information. Experimental results in Section 5 confirm that the COWEXABS model performs better than the state-of-the-art disaster specific summarization models. As a final contribution, we have made the codes and datasets publicly available at https://github.com/krudra/extractive_abstractive_summarization_2019.

2 Related work

Now a days Twitter has evolved as an important source of real-time information during disasters. Real-time information posted by affected people on Twitter helps in improving disaster relief operations [4], [9], [10], [11]. However, we need to extract time-critical situational updates for effective planning by relief organizations [12].

In recent times, researchers have put a lot of effort in summarizing information from microblogging sites like Twitter. Shou et al. [13], [14] used clusters of related tweets and LexRank [15] based extractive summarization technique to summarize evolving tweet stream. In recent times, researchers tried to extract and summarize situational information from Twitter [16], [17], [18], [19]. Nguyen et al. [6] extracted subjects, named entities, events, numerals from tweets, developed a graph among tweets, generated clusters of related tweets, and finally applied PageRank based iterative update scheme within the tweets present in each cluster to get rank of the tweets (TSum4act). A greedy strategy to track real-time events was proposed by Osborne et al. [20]. Recently, Rudra et al. [1] proposed ILP based summarization method COWTS which maximizes the coverage of content words in the summary. In contrast, an extractive disaster-specific summarization method for news articles was proposed by Kedzie et al. [3]. However, summarization of evolving tweet stream poses more challenges than formal news articles and blogs due to the following reasons — (i). tweets provide continuous stream of data evolving over time and therefore real-time processing is a requirement, and (ii). tweets are in general noisy, contain incomplete words, sentences, out-of-vocabulary words [21] and their tone is different from the formal languages used in news articles.

All the above mentioned methods generate summaries that are merely a collection of tweets, i.e., they try to select tweets/sentences based on some criteria (extractive [22] in nature). However, abstractive summarization methods can combine information from related tweets and produce less redundant summary. To this end, a bigram word-graph based abstractive tweet summarization method was proposed by Olariu [23] to handle online stream of tweets in real-time. In the word-graph a node is represented by a bigram but POS-tag information is not considered in the graph. However, this may lead to spurious fusion of tweets because the same bigrams may be used in different context. Furthermore, this is a generalized method and does not consider specific traits of disaster related tweets. Recently, Banerjee et al. [24] proposed an abstractive summarization method on news articles that used word-graph with POS-tag information. New sentences are generated by traversing the word-graph and finally best sentences are selected based on the ILP-based optimization function. The optimization

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

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 our prior work [2], we combined the positive aspects of the above studies - (a) we used a variant of [24] for tweet fusion but introduced an initial extractive step to enable the graph to generate new sentences in real-time, (b) POS-tag information of each bigram was also considered to avoid spurious fusions. (c). disaster-specific content words were also employed to measure the importance of tweets [1]. However, in this paper, we observe that this path generation step has some limitations. Considering bigrams with POS tags prevents spurious fusions but it also reduces the number of paths generated because probability of a bigram appearing in two or more related tweets is much less compared to unigrams.

All the above mentioned summarization methods are unsupervised in nature i.e., they do not need any ground truth summary to train their models. Hence, they can be easily deployed over new datasets. In recent times, researchers have also proposed deep learning model (GRU [25], RNN [26]) based supervised summarization methods for formal news articles. For this, a good amount of dataset and corresponding gold standard summaries are required to train the tweet specific disaster summarization models. However, we observe that vocabularies used in different disasters are significantly different [1] and models trained over past disaster events hardly help in future events [10]. Hence, in this paper, we restrict our focus on real-time unsupervised summarization models.

Most of the disaster specific tweet summarization techniques [1], [2], [6], [27], [28] rely on some particular words i.e., nouns, verbs, numerals, and locations to capture disaster related situational updates. However, they did not consider peculiarities of tweets. In a recent study, Kong et al. [7] showed that a tweet may contain more than one component. In this paper, we observe that singleton components (components containing only one word) contain noises most of the times and they do not play an effective role in the summarization process.

In this work, we keep the positive aspects of our earlier proposed method COWABS [2] and also remove the following limitations to improve our method — (a). earlier we only consider paths; hence, lots of information is lost because sometimes it is not possible to combine information from two tweets to generate a new path. In this work, we consider both raw tweets and new generated paths in the summarization, (b). in COWABS, we consider all the nouns, verbs, and numerals as content words. In the present work, we realize that the singleton components which contain nouns, verbs, numerals are basically noises and we remove them from the list of content words, (c). ILP method of COWABS tried to maximize the coverage of content words but it does not consider importance of different words. In this paper, importance/weight of content words is also taken into account. Details of the methodology will be elaborated subsequently.

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 (≤ 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

3 Dataset and Classification of Messages

We considered following three publicly available disaster datasets shared by Imran et al [29].

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

The datasets are classified into broad humanitarian categories using the AIDR [5] 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 [11]. Around 2,000 messages from each of the three datasets were labeled by the crowdworkers[2], into different classes/categories such as 'infrastructure damage', 'missing or trapped person', 'injured persons', 'shelter needs' etc. These human-labeled messages are used to train AIDR classifier and then it is used to classify subsequent messages in real-time. In this work, we only consider messages for which classifier's confidence score is $\geq$ 0.80. Table 1 shows the categories and detailed data statistics of three disaster events.

4 Summarization

After getting AIDR classified messages with confidence score $\geq$ 0.80 (as described in Section 3), we describe our two step extractive-abstractive summarization approach to generate summaries from each category/class.

For our automatic summarization approach, we consider the following key characteristics:

1) A summary should cover most of the important situational updates, i.e. it should be rich in terms of information coverage.
2) A summary should be less redundant, i.e., it must be able to capture important updates and discard duplicate or near-duplicate information.
3) The summary should be generated in *near real-time*, i.e., we should not overload the summarization

[2] www.crowdflower.com

method with heavy computation such that by the time the summary is produced, the utility of that information is marginal.

We are able to achieve the first two objectives through abstractive summarization and near-duplicate detection. However, maintaining the third constraint (generating summary in near real-time) is difficult. In order to fulfill these objectives, we propose an extractive-abstractive summarization framework. The first phase (extractive phase) uses summarization approach COWTS [1] and selects a subset of tweets that cover most of the information. Next, we run abstractive method over that subset of tweets.

4.1 Extractive Summarization Approach

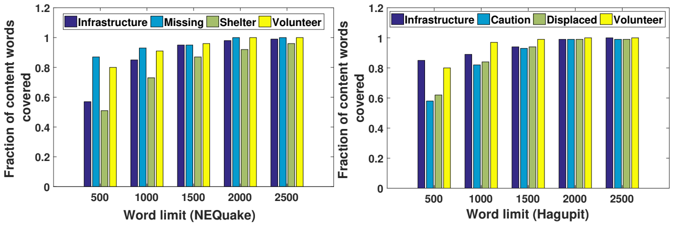
We can use specific traits of disaster-related tweets to construct our extractive summaries.
Content Words: In our prior studies [1], [2], we identified that some specific type of words play a key role in capturing important situational snapshots. Such terms are defined as content words and they are — (i). numerals, (ii). nouns, (iii). location, (iv). main verbs.

Duplicates: Moreover, people post lots of duplicate posts in Twitter during disaster and most of them are redundant. For example, Dharahara tower was collapsed during Nepal earthquake. We observe that this information is communicated in the following five ways:

1) RT @RT_com: #NEPAL: Historic #Dharahara Tower dating back to 1832 reportedly collapses in #Kathmandu [URL]
2) RT @BopsieChroedar: Dharahara Tower Then and Now: A History of Earthquakes in Nepal via @josephjett #India [URL]
3) RT @meghamamgain: The historic #dharahara tower now reduced to a rubble #NepalQuake @ibnlive [URL]
4) RT @AFP: #BREAKING Kathmandu's landmark Dharahara tower collapses after quake: witnesses [URL]
5) RT @Akashtv1: #Nepalquake pulls down landmark buildings - #Dharahara [URL]

In the summarization model, each class is considered (missing or injured people, infrastructure damages, shelter and supplies, $\cdots$) separately and we try to retrieve compact summaries for these classes. Specifically, day-wise snapshots are taken for each class, i.e., a summary of the desired length (number of words) over each day for each of the classes is produced by the system using COWTS [1] in this extractive phase. Duplicate and near-duplicate tweets are removed using the similar technique developed by Tao, et al., [30]. While we remove duplicate tweets from the summarization framework, when we compute the importance of individual words, we also make use of their occurrence in retweets.

The main objective of this phase is to collect most of the content words within small number of tweet set. This stage basically ensures that the next abstractive summarization step is able to generate paths from those tweets and rank them in near real-time. There is a trade-off between the number of words selected in the summary and path generation and ranking time of the next phase. We observe that increment in the summary word length hampers the real-time constraint of the path generation phase. We elaborate this trade-off next.



(a) **NEQuake** (b) **Hagupit**

Fig. 2: Variation in the coverage of content words with number of extracted tweets.

Content-word coverage vis-a-vis length: In Figure 2, we show how the coverage of content words varies with the number of tweets extracted from the whole dataset for different classes of tweets posted during the Nepal earthquake and Typhoon Hagupit. We compute these values for all the three dates and Figure 2 reports the average value for each of the classes across three days. We also observe a similar pattern for the Pakistan flood. As we increase the word length, the summary coverage gradually increases. However, this creates a bottleneck for the next step, i.e., generation of paths from these tweets in near *real-time*. We observe that the running time of the path generation and path ranking phases grows exponentially as the word length of the summary increases. We observed that maintaining the real-time constraint beyond a word length of 1000 is not realistic.

From Figure 2, we can notice that around 1000 word limit is able to capture around 80% content words and number of extracted tweets are also such that abstractive phase (described next) is able to construct paths from these tweets in real time. An informative set of 1,000 words turn out to be sufficient for the next stage of summarization because we consider original tweets along with the generated paths which ensures that there is no information loss. Hence, we decide to produce an initial summary of 1,000 words in the extractive summarization stage.

After extracting a set of informative and important tweets, we focus on preparing a more concise and comprehensive summary through a COntent Words based EXtractive-ABstractive Summarization (**COWEXABS**) approach using these tweets (described next).

4.2 Abstractive Summarization

In this step, our goal is to generate an abstractive summary by combining information from multiple related tweets. In general, the abstractive summaries are more comprehensive than extractive summaries because they contain more information compared to the extractive summaries of the same length (in words). In our proposed summarization method,

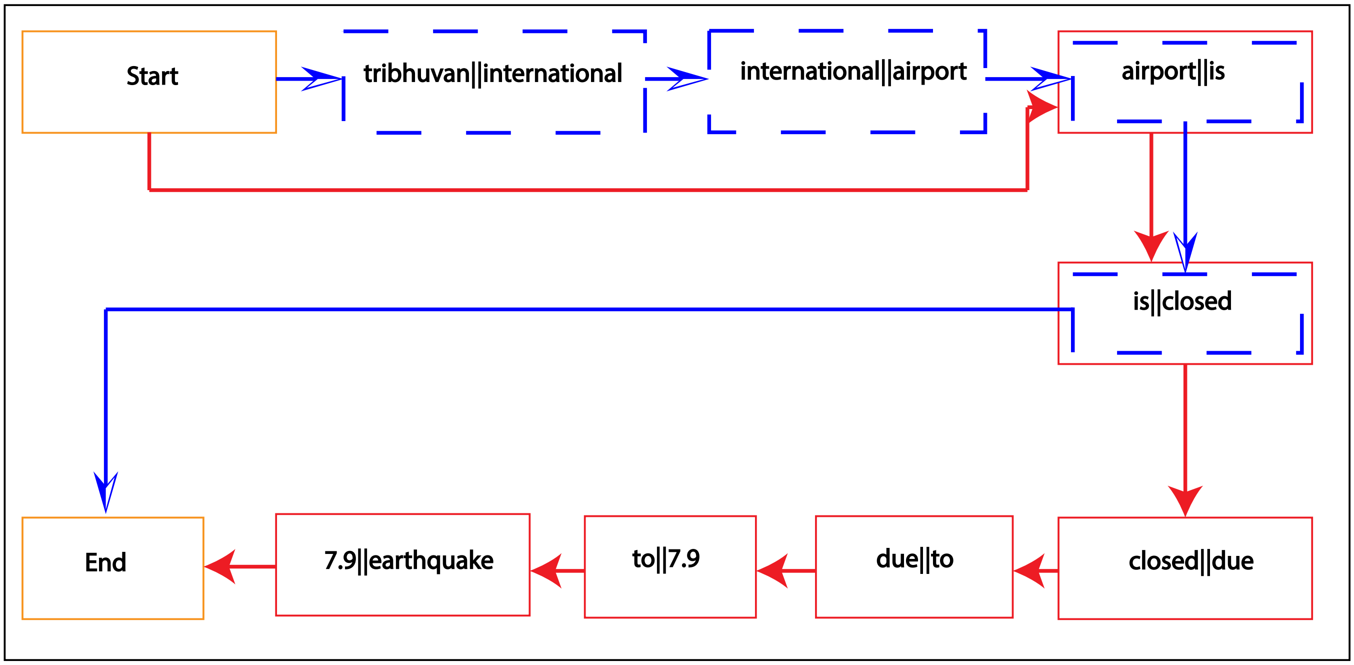


Fig. 3: Bigram word graph generated using following two tweets (1. Tribhuvan international airport is closed, 2. Airport is closed due to 7.9 earthquake) (POS tags are not shown for clarity). Different colors and patterns represent nodes belonging to different tweets. Common nodes contain both the colors. Start and End are special marker nodes.

we have tried to maximize the coverage of informative tweets and remove redundant information jointly. An over-generate and select [31] strategy is followed where a new sentence is generated by combining information from multiple related tweets. Our proposed summarization method tries to generate a summary by selecting important sentences based on two optimization factors: Informativeness, and Redundancy. We have to maximize informativeness and minimize the redundancy in order to make the summary compact and comprehensive. Informativeness is defined as the amount of information in the summary, and we use a centroid-based ranking score to measure the informativeness. We adapt the unigram (with POS tag) based word graph method for path generation proposed by Banerjee, et al. [24] for news articles but several modifications are made to make the system suitable for noisy tweets. We use bigram-based model instead of simple unigram-based model. In bigram-based model, we consider POS tags along with words and this helps to capture the context well. This adaptation helps to improve the grammaticality of generated sentences and avoids generation of spurious sentences by reducing the fusions.

Sentence Generation Process: In order to generate sentences, a word-graph [32] is built with the entire tweet set where we iteratively add each tweet to the graph with the nodes represented by bigrams (adjacent words along with their parts-of-speech (POS) tags[3]). Consecutive words in a sentence represent an edge in the graph. At the time of adding a new tweet to the graph we follow the following steps — (i). the new bigram is merged with an existing node if the words in the bigrams have the same lexical form and POS tag. (ii). In other cases, we create a new node in the graph.

Figure 3 shows an example of our bigram-based word-graph construction, We label each node in the form $w1 \parallel w2$, where first and second word in every bigram is represented by $w1$ and $w2$, respectively. The beginning and end of each tweet is indicated by two special marker nodes *start* and *end* respectively. Our proposed method generates the graph considering the following two tweets that were tweeted on a particular day and AIDR system assigned them to the same infrastructure class — (i) *Tribhuvan international airport is closed*, and (ii) *Airport is closed due to 7.9 earthquake*. We lower-case all words during the graph construction. After formation of the graph, we traverse the paths in the graph between the dummy Start and End nodes to generate the *tweet-paths*. For example, from the graph in Figure 3, we can easily generate a tweet-path such as *Tribhuvan international airport is closed due to 7.9 earthquake*. Within the similar or same word limit, such sentences might contain more information compared to the original tweets. We set a minimum (10 words) and maximum (16 words) length for a sentence to be generated. Such constraints are applied to get rid of very long sentences which might be grammatically ill-formed and very short sentences which are basically noises and do not convey any useful information. In a real-scenario, several thousands tweet-paths can be generated due to multiple points of merging across several tweets.

As reported in Section 1 and 2, bigram based word-graph helps in reducing spurious fusions but on the other hand it also reduces the probability of fusion and formation of new

[3] We use a Twitter specific POS tagger [33] because it is able to identify Twitter-specific tags such as hashtags, mentions, URLs, emoticons along with regular parts-of-speech tags. Such Twitter-specific tags are ignored in path generation step because they are not useful and may hamper readability.

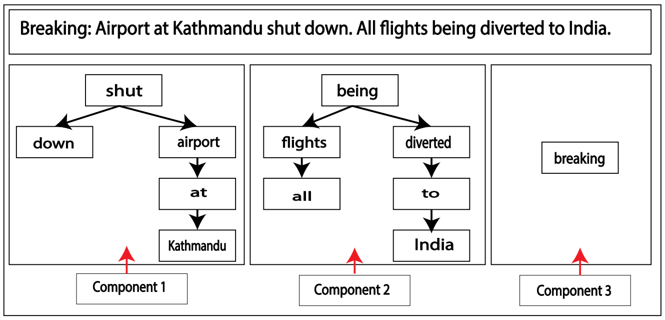


Fig. 4: Dependency trees of multiple components present in a tweet.

paths. Only consideration of paths (not raw tweets) may lead to loss of information in the final summarization step. After this step, we have a set of tweets (extracted in first step) and tweet-paths (generated from the extracted tweets using graph traversal) in our hand and we consider both of them for our final summarization phase. Our goal is to select the best tweets or tweet-paths with the objective of generating a diverse and informative summary. We devise an ILP formulation to select final tweets, tweet-paths and construct the summary.

Removing Singleton Components and Extracting important Content words: While our prior work [1], [2] considered all nouns and verbs as content words, in reality, all the nouns and verbs present in a tweet are *not* related to disaster events. Hence, in the present work, we attempt to identify the key nouns and verbs, and consider only those as *content words*.

To identify key nouns and verbs, we explore the dependency relation among the words in a tweet using a *dependency tree* [7]. A dependency tree basically indicates the relation among different words present in a tweet. For example, dependency tree for the tweet 'flights canceled, evacuations begin in nepal' contains the following four dependency relations – (flights, canceled), (evacuations, begin), (in, begin), (nepal, in). Note that the dependency tree for a particular tweet may contain multiple connected components [7]. For example, the tweet 'Breaking: Airport at Kathmandu shut down. All flights being diverted to India' contains three components as shown in Figure 4. In the third component, 'breaking' is a noun but it is a singleton component and has no effect as a content word. We also observe that many tweets are written in the form 'breaking: 150 feared dead in the quake', 'Update, 10 people killed', consisting of two connected components ('breaking' and '150 feared dead in the quake' for the first one). Such singleton noun components like 'breaking', 'update' are basically noises and do not contribute any effective information to the set of content words. Hence, in this work, we ignore following two kinds of words — (a). which form singleton components, and (b). words in a tweet which are followed by ':' symbol. In the second case, words are used just to promote the importance of the original tweet. After this step, we finally get an important set of *pruned content words*.

ILP Formulation: The ILP-based technique optimizes based upon two factors - (i) weight of the pruned content words (this is similar to that adopted during the extractive phase except singletons): The formulation tries to maximize the number of important pruned content words in the final summary. Importance of a pruned content word is captured through its weights. and (ii) Informativeness of a tweet or tweet-path.

TABLE 2: Notations used in the summarization technique

Notation	Meaning
L	Desired summary length (number of words)
n	Number of *tweets and tweet-paths* considered for summarization (in the time window specified by user)
m	Number of distinct content words included in the n *tweets and tweet-paths*
i	index for *tweets and tweet-paths*
j	index for content words
x_i	indicator variable for *tweets and tweet-paths* i (1 if *tweets or 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 *tweets or tweet-paths* i
$Score(j)$	tf-idf score of content word j
I(i)	Informativeness score of the *tweets or tweet-paths* i
T_j	set of *tweets and tweet-paths* where content word j is present
C_i	set of content words present in *tweets or tweet-paths* i

1) **Weight of the pruned content words** $(Score(j))$**:** TF-IDF score of the content words is computed in the first step (extractive phase) of summarization as proposed in [1]. These weights are also used in this phase as a proxy to determine the importance of the content words.
2) **Informativeness**$(I(i))$**:** Centroid based ranking scheme is used as a proxy to determine sentence importance. Centroid-based ranking [34] tries to capture sentences which are central to the topic of discussion of a document. TF-IDF vector is used to represent each sentence and the mean of the vectors of all the sentences is used to represent the centroid. We measure the cosine similarity value between sentences and compute the centroid. Finally, ILP formulation uses this score as an informative component in the summarization. Importance of a tweet-path is normalized in [0,1] scale. For the original raw tweets, we use machine predicted confidence scores as their informativeness score.

The summarization of L words is obtained by optimizing the following ILP objective function, whereby the highest scoring *tweets and tweet-paths* are returned as output of summarization. The equations are as follow:

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

TABLE 3: Examples of missing person information posted during Nepal earthquake

Smita Magar(28) from Rukumkot, #Nepal, last known she was in Kathmandu Any info would be appreciated.
Last seen at Birjung. Family members trying 2locate Krija (mother)n Piu(child) pl rt @tajinderbagga

subject to the constraints

$$\sum_{i=1}^{n} x_i \cdot Length(i) \leq L \tag{2}$$

$$\sum_{i \in T_j} x_i \geq y_j, j = [1 \cdots m] \tag{3}$$

$$\sum_{j \in C_i} y_j \geq |C_i| \times x_i, i = [1 \cdots n] \tag{4}$$

where the symbols are as explained in Table 2. Both the number of *tweets and tweet-paths* (through the x_i variables) and the number of important content-words (through the y_j variables) are considered by the objective function. Eqn. 2 ensures that the summary length should be at most L, i.e., number of words present in selected tweets and tweet-paths are at most L (user-specified). Eqn. 3 ensures that if objective function selects content word j in the summary, i.e., if $y_j = 1$, then it should select at least one *tweet or tweet-path* containing that content word j. Similarly, Eqn. 4 ensures that all the content words present in a *tweet or tweet-path* i must be included in the summary if *tweet or tweet-path* i is selected for the summary.

We use the GUROBI Optimizer [35] to solve the ILP. After solving this ILP, the set of *tweets and tweet-paths* i such that $x_i = 1$, represent the summary at the current time.

4.3 Missing person summarization

We observe that people post information about their missing friends, relatives, and victims during a disaster scenario. Such information is present in the missing class and is hidden within other general kinds of information like helpline numbers, safety check, launching of Google person finder etc. Ground-level rescue workers need specific details about missing persons like their name, last location, contact number, age etc. to launch search operations. Note that, this is an example of specialized summary; hence a customization of the general framework presented in Eqn. 1 is necessary to produce such summaries.

Such tweets do not contain any content words (nouns, verbs, numerals) and important information is centered around 'name', and 'relation' of missing persons. We observe that path generation is not required for such kind of specific summaries because this kind of information is very sparse and tweets can be hardly combined to produce any new effective sentence. Table 3 shows examples of such tweets. We consider the following parameters as content words for this summarization task:

1) **Name:** name of the missing person[4].
2) **Relation:** personal relations like 'brother', 'wife', 'son', 'friend' etc., as mentioned in the tweet.

The performance of our proposed summarization techniques is discussed in the next section.

5 Experimental Setup and Results

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

5.1 Experimental Settings

Given the machine-classified messages from our datasets: NEQuake, Hagupit, and PFlood, we split 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.

Establishing gold standard summaries: We employed three human volunteers to generate the ground truth summaries. All the volunteers are regular users of Twitter, have good proficiency in English, and are part of the DISARM project which is related to the development of a framework for post-disaster situational analysis and management[5]. Volunteers independently go through all the tweets of a particular class for a particular day and generate a summary of 200 words. Thus, for each information class over each day three human volunteers individually prepared summaries of length 200 words from the tweets. To prepare the final gold standard summary for a particular class, first, we chose tweets, which were unanimously selected by all the volunteers. After that, we considered tweets, which were included by majority of the volunteers until the word limit is crossed. In this way, we create a gold-standard summary containing 200 words for each class.

Baseline approaches: We use four state-of-the-art *unsupervised disaster-specific* 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 proposed by Rudra et al. [1].
2) **COWABS:** is a content word based abstractive summarization approach proposed in our prior work [2].
3) **APSAL:** is an affinity clustering based summarization technique proposed by Kedzie et al. [3].
4) **TSum4act:** is an extractive summarization method proposed by Nguyen *et al.* [6]. It is specifically designed for generating summaries from disaster-related tweets.

Evaluations: We perform two types of evaluations. First, standard ROUGE [36] metric is used to evaluate the performance/quality of summaries produced by our proposed method and baselines (quantitative analysis). We choose F-score of the ROUGE-1 variant only because tweets are in general informal in nature. Formally, ROUGE-1 recall is unigram recall between a candidate / system summary and a reference summary, i.e., how many unigrams of reference summary are present in the candidate summary normalized

[4] We have used the Stanford named-entity-tagger [8] for name detection

[5] https://itra.medialabasia.in/?p=635

by the count of unigrams present in the reference summary. Similarly, ROUGE-1 precision is unigram precision between a candidate summary and a reference summary, i.e., how many unigrams of reference summary are present in the candidate / system summary normalized by the count of unigrams present in the candidate summary. Finally, the F-score is computed as harmonic mean of recall and precision. Along with quantitative analysis, user-studies (qualitative analysis) are also performed using paid crowdsourcing (described below).

5.2 Performance comparison

Evaluation using gold-standard summaries: Table 4 depicts the ROUGE-1 F-scores for the five algorithms for each class and day. We can see that COWEXABS performs better than other baselines in most of the cases (27 out of 30 instances — 90% cases). COWEXABS performs better compared to APSAL and TSum4act by 23% and 26%, respectively, in terms of information coverage (ROUGE-1 score). Combining tweet-paths with raw tweets and removing singleton components helps to increase the coverage over COWTS by 2% to 3%. It is interesting to note that, COWEXABS is performing better compared to COWABS (by 13%) which only considers paths instead of both tweets and paths and do not consider importance/weight of content words in the final summarization stage. We discuss the reasons behind such improvements in the end of this subsection.

Evaluation using crowdsourcing: We use the Figure-Eight[6] crowdsourcing platform to perform the subjective judgment of the generated summaries. We take summaries generated from each class for each day using our proposed method 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 (i.e., one proposed and four from baseline methods) and the three evaluation criteria with their descriptions (as described below). Each of the instances/ tasks[7] is evaluated by ten crowd workers. 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, the Typhoon Hagupit of 2014, and Pakistan flood which happened in 2014. We aim to built an automatic method to generate summaries/reports useful for situational awareness (information that helps understand the situation on the ground after an incident) for crisis responders. For this purpose, we have used five different methods and we want to compare which one is better based on the following criteria: Information coverage, Diversity and comprehension”.

Given the summaries and their topic, We asked three questions to the workers on Figure-Eight as follows:

1) (Q1) Overall, which method in your opinion has the best information coverage?
2) (Q2) Overall, which method provides the most diverse information?
3) (Q3) Overall, which summary helps you quickly understand and comprehend the situation?

We also check the confidence of the annotators before considering their feedback into our result analysis part. This confidence score basically reveals whether they are able to understand the question and judge different summaries appropriately. For information coverage (Q1), diversity (Q2), and comprehension (Q3) part, we get an average confidence score (standard deviation) of 0.72(0.14), 0.68(0.17), and 0.67(0.16) respectively. This indicates that the annotators are more or less confident in the above mentioned evaluation task.

Q1. Information coverage corresponds to the richness of information a summary contains. For instance, we will consider a summary better in terms of information coverage if it contains more crisis-related informative sentences/tweets. From Table 5, we can observe that COWEXABS is able to capture more information compared to other baseline approaches in around 90% cases. This observation is quite consistent with the findings from Table 4. COWEXABS performs better than the other baselines in around 90% cases in terms of ROUGE-1 score.

Q2. Diversity corresponds to the redundancy of tweets in a summary. A good summary should contain diverse/ less redundant set of informative tweets. While we do not use any explicit parameter to control diversity, the ILP framework relies on importance score of the content words, which helps in capturing information from various dimensions. We can see from Table 5 that the proposed summaries are found diverse in around 90% cases.

Q3. Summary understanding attempts to evaluate the easiness in comprehending the summary. In this question, the workers are asked whether they get a mental picture of the situation and can think of some action after reading the summary. From Table 5, it is clear that a large number of respondents found that COWEXABS makes the comprehension task much easier compared to the other baselines. Almost 88% of the summaries produced by COWEXABS are easier to comprehend compared to others.

To give a flavor of the kind of summaries produced by the proposed summarization approach, Table 6 shows summaries generated by COWEXABS 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 the summary returned by COWEXABS is more informative and diverse in nature compared to COWTS (the most competitive baseline). For instance, we can see that the COWEXABS summary contains information about flights, airport updates, damages of buildings, and information sources.

Time taken for summarization: As stated earlier, our primary objective is to generate the summaries in near real-time. Hence, we analyze the execution times of COWEXABS and baseline methods. Table 7 provides detailed information about run-time of our proposed COWEXABS method[8] and four other baselines. APSAL requires more time over large

[6] https://figure-eight.com/

[7] Terms instances and tasks are used interchangeably in this paper.

[8] For COWEXABS we consider the time taken to generate the dependency parse tree (required in order to remove singleton components) and producing the final summary.

TABLE 4: Comparison of the ROUGE-1 F-scores (with classification, twitter specific tags, emoticons, hashtags, mentions, urls, removed and standard rouge stemming(-m) and stopwords(-s) option) for COWEXABS (the proposed methodology) and the three baseline methods (COWTS, COWABS, APSAL, and TSum4act) on the same situational tweet stream for each class, for each day, and for each dataset.

Step size	**ROUGE-1 F-score (NEQuake)**														
	Infrastructure					Missing					Shelter				
	COWEXABS	COWTS	COWABS	APSAL	TSum4act	COWEXABS	COWTS	COWABS	APSAL	TSum4act	COWEXABS	COWTS	COWABS	APSAL	TSum4act
25/04/2015	**0.5190**	0.4842	0.3866	0.3691	0.3758	**0.5468**	0.5353	0.3082	0.3162	0.1901	**0.5165**	**0.5165**	0.4513	0.4548	0.4742
26/04/2015	0.3323	**0.3496**	**0.3496**	0.3071	0.2387	**0.3806**	0.3066	0.3034	0.3496	0.3694	**0.3693**	0.3674	0.3387	0.3275	0.3610
27/04/2015	**0.3861**	0.3631	0.3352	0.3657	0.3765	**0.3643**	0.3494	0.3275	0.3478	0.2825	**0.4371**	0.4340	0.3922	0.3238	0.3631
Step size	**ROUGE-1 F-score (Hagupit)**														
	Infrastructure					Caution					Displaced				
	COWEXABS	COWTS	COWABS	APSAL	TSum4act	COWEXABS	COWTS	COWABS	APSAL	TSum4act	COWEXABS	COWTS	COWABS	APSAL	TSum4act
06/12/2014	0.5529	**0.6190**	0.5364	0.4946	0.5655	**0.4498**	**0.4498**	0.4259	0.2922	0.3566	**0.4309**	0.3955	0.3676	0.2881	0.2558
07/12/2014	**0.6114**	**0.6114**	0.4702	0.4339	0.4852	**0.3423**	0.3303	0.3333	0.3202	0.3281	**0.3585**	**0.3585**	0.2905	0.2500	0.2307
08/12/2014	**0.4800**	**0.4800**	0.4637	0.3891	0.4413	**0.4217**	0.4169	0.3147	0.3803	0.4125	**0.4652**	0.4277	0.4144	0.3376	0.3812
Step size	**ROUGE-1 F-score (PFlood)**														
	Infrastructure					Missing					Volunteer				
	COWEXABS	COWTS	COWABS	APSAL	TSum4act	COWEXABS	COWTS	COWABS	APSAL	TSum4act	COWEXABS	COWTS	COWABS	APSAL	TSum4act
07/09/2014	0.6593	**0.7232**	0.6762	0.6894	0.7191	**0.5935**	**0.5935**	0.5705	0.5787	0.5769	**0.3591**	0.3378	0.3419	0.2646	0.2092
08/09/2014	**0.7258**	0.7235	0.6926	0.6781	0.6315	**0.4898**	0.4758	0.4436	0.4705	0.4498	**0.3218**	0.2865	0.3207	0.2105	0.2631

TABLE 5: Results of the crowdsourcing based evaluation of the system summaries for COWEXABS (our proposed methodology) and the four baseline techniques (COWTS, COWABS, APSAL, TSum4act). Values in the table indicate fraction of instances where our proposed method is preferred over other baselines for a particular question.

Datasets	**Method**	**Q1**	**Q2**	**Q3**
NEQuake	COWEXABS	1	1	1
	COWTS	0	0	0
	COWABS	0	0	0
	APSAL	0	0	0
	TSum4act	0	0	0
Hagupit	COWEXABS	0.92	0.92	0.83
	COWTS	0	0	0
	COWABS	0.08	0	0.17
	APSAL	0	0.08	0
	TSum4act	0	0	0
PFlood	COWEXABS	0.83	0.83	0.83
	COWTS	0	0	0.17
	COWABS	0	0	0
	APSAL	0.17	0.17	0
	TSum4act	0	0	0

TABLE 6: Summary of length 50 words (excluding #,@,RT,URLs), generated from the situational tweets of the infrastructure class (26th April) by (i) COWEXABS (proposed methodology), (ii) COWTS.

Summary by COWEXABS	**Summary by COWTS**
RT @cnnbrk: Nepal quake photos show historic buildings reduced to rubble as survivor search continues http://t.co/idVakR2QOT. Reporter: Kathmandu Airport closed following 6.7 aftershock; no planes allowed to land - @NepalQuake https://t.co/Vvbs2V9XTX. #NepalEarthquake update: Flight operation starts from Tribhuvan International Airport, Kathmandu. Pakistan Army Rescue Team comprising doctors, engineers & rescue workers shortly after arrival at #Kathmandu Airport http://t.co/6Cf8bgeort	#PM chairs Follow-up meeting to #review situation following #earthquake in #Nepal @PMOlndia #nepalquake #NepalQuake. RT @cnnbrk: #Nepal #quake photos show historic buildings reduced to rubble as survivor search continues. http://t.co/idVakR2QOT http://t.co/Z. Pakistan Army rescue team comprising #doctors, #engineers & #rescue #workers shortly arrive at #Kathmandu Airport http://t.co/6Cf8bgeort. @SushmaSwaraj @MEAcontrol-room Plz open HelpDesk at Kathmandu airport. @Suvasit Thanks for #airport #update.

TABLE 7: Runtime (seconds) of different algorithms for each of the classes averaged over three days.

Datasets	**Class**	**COWEXABS**	**COWTS**	**COWABS**	**APSAL**	**TSum4act**
NEQuake	infrastructure	132.41	12.88	21.56	1719.79	16.79K
	missing	105.76	7.20	21.24	646.18	7.97K
	shelter	230.70	16.78	29.51	2685.67	21.45K
	volunteer	21.38	1.98	9.66	10.35	0.84K
Hagupit	infrastructure	65.71	3.02	11.02	57.50	2.01K
	caution	210.97	19.91	28.15	3846.34	33.30K
	displaced	155.35	17.06	31.14	2144.39	22.22K
	volunteer	40.86	4.07	17.03	103.67	2.70K
PFlood	infrastructure	12.32	1.82	8.60	11.37	0.78K
	missing	44.32	3.61	18.44	100.13	2.55K
	volunteer	394.78	56.02	62.15	11542.43	75.69K

TABLE 8: Effect of generated paths and pruned content words on summarization

Datasets	**Class**	**COWEXABS**	**COWEXABS - paths**	**COWEXABS - pruned words**
NEQuake	infrastructure	**0.4124**	**0.4124**	0.4073
	missing	**0.4305**	0.4205	0.3989
	shelter	**0.4409**	0.4357	0.4393
	volunteer	**0.5864**	0.5803	0.5675
Hagupit	infrastructure	0.5481	0.5471	**0.5763**
	caution	**0.4046**	**0.4046**	0.3994
	displaced	**0.4182**	0.4086	0.3927
	volunteer	**0.4497**	**0.4497**	0.4483
PFlood	infrastructure	0.6925	0.6888	**0.7169**
	missing	**0.5416**	**0.5416**	**0.5416**
	volunteer	**0.3404**	0.3326	0.3058

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*, *extraction of events from tweets*, etc. COWEXABS has a higher running time compared to COWTS [1] and COWABS [2] - the time mainly is taken to identify singleton components. However, the algorithm still can be considered as *near real-time* as typically a summary would be produced (say) after an hour.

Evaluating importance of individual parameters: As mentioned earlier, performance of our proposed summarization method COWEXABS depends on two parameters — (i). generated paths, and (ii). removal of singleton components. From Table 4, we can see that COWEXABS performs better compared to other baselines in most of the cases (≥ 80%). In this part, we analyze the contribution of individual parameters in the output of COWEXABS. Table 8 compares the F-scores (averaged over different dates) of COWEXABS in the absence of one of the parameters (path or pruned components). The results show that both the parameters contribute to the quality of the generated summary. and removing any one of them hampers the overall performance. A closer look reveals that pruned content words have more impact than the generated paths in the quality of the generated summaries.

It is also evident from Table 8 that if we consider paths along with original tweets (COWEXABS - pruned words) in the ILP framework, it will perform better than COWTS which only considers raw tweets. In fact, the performance is better than COWTS by 1-2%.

Reason behind better performance: We try to analyze the four baseline algorithms and identify their limitations and thus understand the reason behind the superior performance of COWEXABS. TSum4act prepares clusters of related tweets, applies PageRank over each of the clusters

TABLE 9: Examples of tweets which contain wrongly identified singleton components (marked in red) by the Twitter Parser

SG45, our second DEL-KTM flight today circling at Nepal border awaiting landing clearance from KTM, airport bays full.
Smita Magar(28) from Rukumkot, #Nepal, last known she was in Kathmandu Any info would be appreciated.
.@MSF is also sending 3000 kits of non-food items and medical kits to those affected by the #earthquake in #Nepal.
LIVE updates: #Kathmandu airport closed due to heavy rain, thunderstorm.

to rank the tweets and finally selects one top ranked tweet from each of the clusters. Basically, TSum4act assumes that each cluster is equally important and selects one tweet from each cluster. However, we observe that this does not hold for disasters where some clusters are more important and selecting more than one tweet may be necessary to produce more informative summary. APSAL also maintains clusters of related tweets like TSum4act but it also captures salient score of tweets and ranks cluster centroids based on that score. It slightly overcomes the issues of TSum4act. However, this method also can not pick more than one tweet from a cluster and it is originally proposed for news articles. Hence, some of the features such as *heading of the article*, *sentence position in the document* are not available for tweets which affects the performance of the method. Among all the baseline methods, COWTS shows the best performance in terms of ROUGE-1 scores perhaps due to its simplicity and ability to capture disaster-specific informative words. However, it is an extractive method and it also suffers from standard redundancy problem. Sometimes, two different tweets might contain partially overlapping information but we have to keep both of them in the summary. Side by side, it does not remove the singleton content words which are basically noises. On the other hand, only considering paths (COWABS) does not provide satisfactory result. We have found two reasons behind that — (i). in the path formation step, we build the word graph using bigrams as nodes in order to remove spurious fusions. However, this also reduces the probability of path formation, i.e., combining information from multiple tweets. Hence, if we only rely on paths in the final ILP summarization method (Eqn. 1) then lot of information is lost due to the path formation step. (ii). COWABS maximizes the coverage but it does not consider the importance of content words. In our current model COWEXABS, we have tried to overcome the existing limitations of the earlier models.

Limitations of the model: In Table 4, we observe that the performance of COWEXABS is worse than the baselines in around 10% cases. In this section, we analyze the reason behind such performance drop. In our framework, we make a uniform assumption that all the singleton components are noisy and do not contain any useful information about the situation. However, there exist following issues with this simple assumption — (i). Some of the words are wrongly marked as singleton components by the Twitter parser, and (ii). Informal writing pattern of tweets also poses a problem to the detection of noisy singleton components. For example, users use hashtags for normal terms, unnecessary punctuation marks etc., which also affect the accuracy of the parser. Table 9 shows examples of tweets where informative words are marked as singleton components. Hence, it is evident that discrimination between informative and noisy singleton components is a non-trivial task. Accuracy of this step helps in identifying good set of content words which is helpful for generating more informative situational summaries.

5.3 Performance of missing person information

Since other methods do not provide such specialized summarization, we concentrate on finding its coverage vis-a-vis the produced ground truth.

Establishing gold standard summaries: The ground-truth generation is a bit different than the previous cases because the required kind of information is very sparse. Hence we do not put any restriction on the number of words while generating a gold standard summary; the tweets which pass unanimous judgement from all the (three) volunteers are considered. For three days (25th, 26th, and 27th April), we have created summaries of 30, 305, and 130 words, respectively for the NEQuake event reflecting the availability. Similarly, for the PFlood event, we have created summaries of 110 and 80 words for 7th and 8th September, respectively. Our system also generates summaries of the same length as the ground truth.

Evaluation: Since we are primarily interested in coverage/recall score, we consider the recall of the ROUGE-1 variant only. We have obtained 100%, 82%, 87% score over three days (25th, 26th, 27th) respectively. For 26th and 27th, our proposed method fails to cover some information about missing persons. In case of PFlood, we have obtained 81%, 83% recall score for 7th and 8th September, respectively. The mistakes specially occur where instead of name - only relationship information is present (25%) - like *My brother is missing*. Also, there are mistakes in spelling or use of short-hand expressions (doughter, bro etc.) which our system fails to capture.

6 Conclusion

A large number of tweets are posted during disasters and emergencies. A concise and categorical representation of those tweets is necessary to enable humanitarian organizations for rapid disaster response. In this paper, we propose an approach to generate summaries in near real-time from the incoming stream of tweets. We consider peculiarities of short informal tweets such as presence of singleton components in tweets (noise) and presence of similar information in multiple related tweets (redundancy). Finally, we develop an ILP-based summarization technique to generate a concise report of an event. Specifically, tweets from three natural disasters are used to generate comprehensive abstractive summaries for important humanitarian classes such as infrastructure damage, missing or found people, and shelter needs of the affected people. Results show that removing noisy components and combining information from related tweets helps in better information coverage which satisfies the objective of this work. Furthermore, we realize that information needs of different stakeholders (government, news reporters, rescue personnel, etc.) vary a lot during disasters and some of the facts such as information about persons missing or trapped under buildings is time-critical in nature. In this paper, we try to process those information separately from the generic summarization.

In the summarization phase, we only focus on class-based summarization but we observe that most of the classes contain information from different dimensions. For example, infrastructure class contains information about airport, road, building, etc. In future, we will try to introduce a budget across all these small scale dimensions to improve the coverage and diversity of the summarization method. Side by side, we realize that discrimination between noisy and informative singleton components is a challenging task and more sophisticated methods are required for that. In this paper, we propose a summarization method to get a concise snapshot of the situational information about a given disaster event. However, there exist many other associated challenges such as automatic detection of a disaster event, relevant tweet collection, etc. In future, we will try to combine these two modules with the present summarization scheme to make it a deployable system.

Acknowledgement: This work is an extended version of the paper: Rudra *et al.*, "Summarizing Situational Tweets in Crisis Scenario," Proceedings of The ACM International Conference on Hypertext and Social Media, 2016.

References

[1] 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.

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

[3] 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)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1608–1617. [Online]. Available: http://www.aclweb.org/anthology/P15-1155

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

[5] 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*. International World Wide Web Conferences Steering Committee, 2014, pp. 159–162.

[6] M.-T. Nguyen, A. Kitamoto, and T.-T. Nguyen, *TSum4act: A Framework for Retrieving and Summarizing Actionable Tweets During a Disaster for Reaction*. Springer International Publishing, 2015, pp. 64–75.

[7] L. Kong, N. Schneider, S. Swayamdipta, A. Bhatia, C. Dyer, and N. A. Smith, "A Dependency Parser for Tweets," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2014, pp. 1001–1012.

[8] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning, "Named entity recognition with character-level models," in *Proc. HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 180–183.

[9] H. Gao, G. Barbier, and R. Goolsby, "Harnessing the crowdsourcing power of social media for disaster relief," *Intelligent Systems, IEEE*, vol. 26, no. 3, pp. 10–14, 2011.

[10] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: a survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 67, 2015.

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

[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*. Springer, 2012, pp. 462–469.

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

[14] 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*, vol. 27, pp. 1301–1314, 2015.

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

[16] 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)*, 2010, pp. 851–860.

[17] 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.

[18] 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.

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

[20] 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.

[21] A. Hannak, E. Anderson, L. F. Barrett, S. Lehmann, A. Mislove, and M. Riedewald, "Tweetin' in the Rain: Exploring societal-scale effects of weather on mood," in *Proceedings of the 6th International Conference on Weblogs and Social Media (ICWSM)*. AAAI, 2012, pp. 479–482.

[22] V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 3, pp. 258–268, 2010.

[23] A. Olariu, "Efficient online summarization of microblogging streams," in *Proc. EACL*, 2014, pp. 236–240.

[24] 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.

[25] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 654–663.

[26] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 3075–3081.

[27] K. Rudra, P. Goyal, N. Ganguly, P. Mitra, and M. Imran, "Identifying sub-events and summarizing disaster-related information from microblogs," in *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*, ser. SIGIR '18, 2018, pp. 265–274.

[28] K. Rudra, N. Ganguly, P. Goyal, and S. Ghosh, "Extracting and summarizing situational information from the twitter social media during disasters," *ACM Trans. Web*, vol. 12, no. 3, pp. 17:1–17:35, Jul. 2018.

[29] M. Imran, P. Mitra, and C. Castillo, "Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages," in *Proc. LREC*, 2016.

[30] 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)*, 2013, pp. 1273–1284.

[31] 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. Association for Computational Linguistics, 2001, pp. 1–8.

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

[33] 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.

[34] D. R. Radev, H. Jing, M. Styś, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management*, vol. 40, no. 6, pp. 919–938, 2004.

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

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