Going Beyond Content Richness: Verified Information Aware Summarization of Crisis-Related Microblogs

Anonymous Author(s)

ABSTRACT

High-impact catastrophic events (bomb attacks, shootings) trigger posting of large volume of information on social media platforms such as Twitter. Recent works have proposed content-aware systems for summarizing this information, thereby facilitating post-disaster services. However, a significant proportion of the posted content is *unverified*, which restricts the practical usage of the existing summarization systems. In this paper, we work on the novel task of generating *verified summaries* of information posted on Twitter during disasters. We first jointly learn representations of content-classes and expression-classes of tweets posted during disasters using a novel LDA-based generative model. These representations of content & expression classes are used in conjunction with pre-disaster user behavior and temporal signals (replies) for training a Tree-LSTM based tweet-verification model. The model infers tweet verification probabilities which are used, besides information content of tweets, in an Integer Linear Programming (ILP) framework for generating the desired verified summaries. The summaries are fine-tuned using the class information of the tweets as obtained from the LDA-based generative model. Extensive experiments are performed on a publicly-available labeled dataset of man-made disasters which demonstrate the effectiveness of our tweet-verification (3-13% gain over baselines) and summarization (12-48% gain in verified content proportion, 8-13% gain in ROUGE-score over state-of-the-art) systems.

KEYWORDS

Unverified Information, Twitter, Disaster, Summarization

1 INTRODUCTION

Over the past decade, social networking platforms such as Twitter have become important sources of real-time information especially during high-impact catastrophic events such as bomb attacks, and shootings. Recent researches have shown the potential of utilizing the boundless data accessible in facilitating post-disaster services [37, 43]. Researchers have proposed robust systems for increasing the situational awareness during disasters, typically by generating event-summaries of posts published on Twitter; the systems focus on maximizing situational content in the summaries [4, 30, 36, 37]. The effectiveness of these systems, however, gets severely restricted by the significant proportion of *false & unverified tweets* posted on Twitter besides the true and trustworthy facts [5, 16]; the situation is worse in case of man-made disasters due to its easily exploitable psycho-social impacts on the masses (panic, stress, mental trauma). Furthermore, the unverified tweets (which may subsequently turn out to be false) are sometimes, unintentionally propagated by popular personalities (politicians, celebrities) resulting in their noteworthy attention among the masses [22].

The summary of the situation during a crisis event is required in real time so that the respective authorities/stake holders can take immediate action — this is in essence while various information are still emerging and many of the tweets are unsubstantiated. A verification-aware summarization system, thus, has to take a call on the authenticity of the individual tweets with limited secondary data to verify it. Considering the hardness of the problem, none of the existing summarization systems [4, 20, 30, 31, 36, 37] explicitly make any attempt to minimize the unverified information in the summary; they only concentrate on increasing the richness of content. In this paper, we propose a novel but simple pipeline to generate *verified summaries*; we compute the probability of the tweet being verified (we term it as *verification score*) and then jointly exploit the information content & verification score for generating summaries.

For computing the verification score, we train a Tree-LSTM based architecture which can elegantly model the phenomenon of a tweet being published and several replies/counter replies being posted as a reaction. The model takes as input user's pre-disaster behavior, information about the content class of a tweet (and its reply), the manner in which the tweet has been expressed - this information is efficiently encapsulated using a novel LDA-based generative process. Note that the task of computing verification score for each tweet has few similarities with that of fake-news / fake-event detection [26, 35, 46]; however there are certain important differences. Fake news detection systems usually predict the credibility of a very specific piece of news which is being discussed by a large number of users; the system thus has a lot of signals with them to work with (typically 500-1000 tweets per news). Whereas due to a wide variety of news developing during a disaster, the signals for many of the tweets are inadequate mainly due to limited discussions surrounding them. Therefore, in our model, we put emphasis on exploiting the linguistic and behavioral dynamics of tweets/users for determining the authenticity of tweets (§4). This helps us in performing much better than state-of-the-art fake-news detection algorithms; our model beats such baselines by 3-13% in terms of F1-Score on a publicly available and expert-curated labeled dataset of four man-made disaster events [52] (§6).
Further to this, we use an integer linear programming (ILP) framework for generating summary of a crisis from the tweets posted during the event. We make use of both, information content & verification scores of tweets as optimization parameters to generate a high quality summary (§5). We perform a detailed, careful study of the output generated at various steps which helps us in iteratively fixing the weights of various components of our framework. The generated summaries of the four man-made disasters have exceptionally high proportion of verified content (12-48% gain over state-of-the-art) while still being able to maintain high rouge scores & content richness (§7). Qualitatively analyzing these summaries helps us in understanding the robustness of our framework; the robustness is also evident in a case study of the 2019 Sri Lankan Attacks as we examine in §7.4.

2 RELATED WORKS

Detecting if a disaster-related information is true, false or unverified is a relatively easy task if performed a long time after the disaster. The amount of supporting data accessible from news articles and related webpages can directly provide this knowledge about the information. Unfortunately, these articles are unavailable at the time when disaster-related tweets are emerging. The summary of a disaster needs to be periodically updated with the newly available information and cannot wait for the articles which verify that information. This makes the task of tweet verification challenging. In this section, we discuss works on the tweet verification task and the systems summarizing information posted during crisis.

Handling unverified information: The potential of social media platforms - to be used as a source of creating and spreading unverified information - has triggered a lot of work on its analysis, detection and verification [9, 13, 29]. Most of the unverified tweet detection research is focused around designing hand-crafted features from tweets such as tweet and user features [6], locations [48], multimedia [41]. Some of the other approaches use public opinion, belief identification [25], regular expression [51], temporal pattern of tweet [23, 27], misinformation cascades [12, 15, 28]. Deep learning models (RNN) are also explored to capture verification signals [8]. More recently, researchers have analyzed unverified information posted during disasters [1, 45, 49, 50]. Zeng et al. [50] proposed a classifier to predict the stances to unverified messages (affirmation/denial) posted during disasters. Affirmative tweets have to wait for a longer time to get retweeted by other users in contrast to denial tweets [49]. Starbird et al [40] analyzed the role of journalists in posting and correcting unverified messages during crisis events. The focus of our work is on creating a disaster-specific tweet verification model and using it for generating verified summaries of tweets posted during man-made disasters. Our analyses show that, in the context of disasters, static twitter attributes are not helpful to the tweet verification task (§4.2). Moreover, the work of Zeng et al. [50] suggests that during disasters, the number of affirmative replies to a false tweet are greater than denial replies, thus proving insufficient for the verification task. Hence, additional signals are required and those signals come in the form of content-classes of man-made disasters and the ways of expression of tweets. *Note that both these signals have not been explored by any of the prior works*; the works have been limited to using standard word embeddings (typically word2vec) which don't provide control over differentiating between the two signals. Our model jointly incorporates the content-class of a tweet & its way of expression, dynamically derives user-attributes using her pre-disaster behavior and also utilizes tree-like structure which the replies have for capturing affirmation/denial.

Our work is different from the tasks of fake-news detection [26, 35, 46] and fact-checking [44, 47], both in context & the proposed challenges. Fake news detection systems usually work on a group of tweets related to a particular piece of potentially false news/event (eg. a celebrity getting married); the number of tweets for each event is usually large which provides these systems with a lot of signals to work with. A wide variety of news and a significant number of sub-events [38] develop during a disaster, signals for many of which are inadequate. Nevertheless, we have evaluated state-of-the-art fake news detection methods [26, 35] and shown that they do not adapt well in the not-so-popular (tweet) and minimal supporting evidence scenario. We will later discuss the specific results (Table 4). Likewise, fact-checking systems operate on public verbal statements of politicians & celebrities which is not in the scope of our work.

Table 1: Statistics of the dataset. We divide the dataset into the set of source tweets and replies.

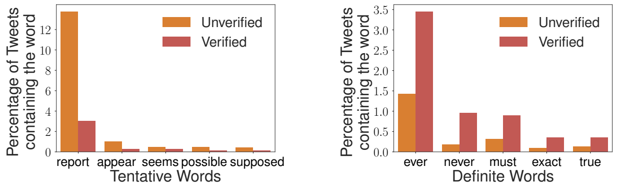
Disaster	# of Tweets	# of Users	# of Source Tweets		
			Verified	Unverified	Total
Charliehebdo	38246	18726	1740	484	2224
Germanwings	4486	2932	232	238	470
Ottawa	12281	7800	424	486	910
Sydney	23989	12224	704	531	1235
Total	79002	41682	3100	1739	4839

Tweet summarization: Several efforts have been made by researchers for generating summaries of large tweet streams [7, 19, 21, 39]. For the task of disaster-specific tweet-stream summarization, researchers have worked on maximizing situational [37], actionable [32], salient [20] and sub-event [38] content in their summaries. However, all these systems work on an inherent assumption that the posts being used by them are verified and trustworthy. In this paper, for the first time, we make use of the verified/unverified knowledge of posts in a disaster-specific tweet-stream summarization setting. The existing summarization systems optimize only on content proportion; the proposed **verification-summarization** system attempts to simultaneously optimize on content & verified proportions, allowing end users to retrieve verified & content-rich summaries of tweet streams.

3 DATASET

We use the dataset created by Zubiaga et al [52] and focus on the tweets from the following four human triggered disaster events : **(i) Charlie Hebdo Shooting (Jan'15):** Shootings involving killing of 11 people in the offices of the French newspaper, Charlie Hebdo, in Paris, **(ii) Germanwings Plane Crash (Mar'15):** Deliberate crashing of a passenger plane by a co-pilot in the French Alps, **(iii) Ottawa Shooting (Oct'14):** Shootings which occurred at Parliament Hill in Ottawa, and **(iv) Sydney Siege (Dec'14):** 10 customers and 8 employees of a Lindt chocolate cafe held hostage in Sydney. The dataset consists of highly-retweeted tweets from these disaster events. We divide the tweets in the dataset into 2 parts: source tweets and replies to the source tweets - relevant tags available in the dataset allow us to do so. Each source tweet in the dataset was labeled either *verified* or *unverified* by a team of journalists as per the scheme developed in [53]; the annotation scheme required the journalists to analyze tweets and all of their replies on certain parameters (supporting evidence, certainty, type of replies). We note that the entire annotation process is complex; it is time consuming and requires domain expertise and thus, it was not prudent for us to create a new dataset. Rather, we reused the dataset which should allow the research community to easily compare against our results.

We consider the annotations of Zubiaga et al as gold standard. Note that this gold standard was developed while the disaster events were either still in progress or had just finished. In §7.3, we relook into a part of the unverified tweets to check if some of them have later been verified. Table 1 contains the statistics.



(a) Tentative words distribution **(b) Definite words distribution**

Figure 1: Percentage of tweets (unverified/verified) containing tentative/definite words. Unverified tweets contain a lot of tentative words. Verified tweets contain definite words.

4 TWEET VERIFICATION MODULE

Our tweet verification module infers a verification score corresponding to a given tweet. It is driven by the following three hypothesis: **(a).** Tweets posted during disasters differ significantly in the way they are *expressed* across the verified and unverified categories. The actual quantum of difference varies according to the *type of content* they convey, **(b).** User's *pre-disaster tweeting behavior* is indicative of her verified information posting tendency, and **(c).** The set of replies to a tweet provide us *temporal signals*, valuable for decoding the truth value of the tweet.

We first learn a latent space jointly capturing the type of content posted and the way in which that content is expressed during disasters (§4.1). To capture the pre-disaster behavior of users (§4.2), we define a metric (*regularity score*). Finally, the tweet and its set of replies are modeled using Tree-based LSTMs for learning the tweet verification task (§4.3).

4.1 Content-Expression Topic Model

Earlier studies [17, 18] have identified content classes for tweets posted during natural disasters (earthquake, flood) — 'infrastructure damage', 'shelter & service', etc. Similar to natural disasters, we expect the man-made catastrophic events to attract tweets belonging to a finite number of *content classes*. Moreover, in the context of tweet verification, the way in which the content is presented and communicated is likely to determine the authenticity of the content. For example, if a tweet is tentatively structured, it is indicative of it being unverified. Figure 1 shows the distribution of a few tentative and definite words across verified & unverified classes. The figure shows that unverified tweets are more prone to use tentative words while verified tweets are more prone to use definite words (Differences are statistically significant — Welch's t-test, $p < 0.05$).

In order to exploit these differences between verified and unverified tweets in content and their means of expression, we learn a unique latent space which jointly captures both these aspects and for learning this latent space, we design a novel LDA-based generative model — **Content-Expression Topic Model (CETM)**. We assume that, the major topics of interest during disasters revolve around a set of *content words* [37]. Content words constitute the terms which convey the key information present in the tweet. They comprise of — (i) **Nouns**: eg. police, terrorists, etc., (ii) **Verbs**: eg. killed, injured, etc., and (iii) **Numerals**. The set of content words in the case of disasters (similar across disasters & limited in number) are different in their characteristics when compared to generic events (vary across events & linearly growing), thus making their generalization easier [37]. Furthermore, we also want these topics to capture *expression words* present in the tweets. Expression words comprise of terms which depict the following psycho-linguistic characteristics — (i) **Tentativeness:** eg. probably, reported, maybe, (ii) **Certainty:** eg. confirmed, assure, must, (iii) **Negation:** eg. can't, isn't, neither, and (iv) **Enquiring:** eg. how, what, why, etc. They enable us to model the mechanism in which both the source tweets (tentative or certain) and the replies (enquiries or denial) are phrased. We present the details of CETM next.

4.1.1 *Generative process of CETM:.*

Let $\mathcal{T}$ be the set of tweets, C_v be the content-word set, $\mathcal{E}_v$ be the expression-word set and $\mathcal{J}$ be the set of tweet categories (Tweet/Reply). We define each tweet $t_i \in \mathcal{T}$ as a tuple $t_i = (W_i^{(c)}, W_i^{(e)}, j_i)$ where $W_i^{(c)} \subseteq C_v$, $W_i^{(e)} \subseteq \mathcal{E}_v$, $j_i \in \mathcal{J}$. While the motivation of using content words and expression words follows directly from our observations, using the tweet category helps us in distinguishing the source tweets from replies. Let $\mathcal{K}^{(c)}$ be the set of *content topics* (describing content classes) and $\mathcal{K}^{(e)}$ be the set of *expression topics* (describing the communication characteristics - expression & tweet category). A user who wants to post a tweet first chooses a content topic k_c, and then selects an expression topic k_e under k_c to determine her communication mechanism for the content. The user then generates the set of content words $W_i^{(c)}$ from the chosen content topic and the set of expression words $W_i^{(e)}$ & the tweet category j_i from the chosen expression topic.

4.1.2 *Inferring CETM's parameters:*

We use a collapsed Gibbs sampling approach for inferring the model parameters. The likelihood of generating t_i from a content topic k_c is given by:

$$p(W_i^{(c)}|k_c) \propto \frac{n_{k_c} + \alpha_{k_c}}{|\mathcal{T}| + |\mathcal{K}^{(c)}|\alpha_{k_c}} * \prod_{w \in W_i^{(c)}} \frac{n_{k_c}^{(w)} + \alpha_{w_c}}{n_{k_c}^{(.)} + |C_v|\alpha_{w_c}} \tag{1}$$

And, the likelihood of generating t_i from an expression topic k_e is:

$$p(W_i^{(e)}, j_i|k_e) \propto \frac{n_{k_e} + \alpha_{k_e}}{|\mathcal{T}| + |\mathcal{K}^{(e)}|\alpha_{k_e}} * \frac{n_{k_e}^{(j_i)} + \alpha_j}{n_{k_e}^{(.)} + |\mathcal{J}|\alpha_j} * \prod_{w \in W_i^{(e)}} \frac{n_{k_e}^{(w)} + \alpha_{w_e}}{n_{k_e}^{(.)} + |\mathcal{E}_v|\alpha_{w_e}} \tag{2}$$

where n_{k_c}, n_{k_e} are the counts of tweets assigned to topics k_c and k_e respectively. $n_{k_c}^{(w)}$, $n_{k_e}^{(w)}$ are the number of times word w was assigned to topic k_c, k_e respectively and $n_{k_e}^{(j_i)}$ is the number of times j_i was assigned to topic k_e. $n^{(.)}$ denote the respective marginals.

4.1.3 *Effectiveness of CETM:.*

We check if CETM is able to learn distinctive topics encapsulating the desired content classes and their ways of expression. We use Twitie [3] for POS tagging tweets and utilize the POS tags for extracting content words. We obtain word lists of four psycho-linguistic characteristics using LIWC [33]. The words of a tweet are searched in these lists for extracting expression words. We initialize the Dirichlet priors using the well-established strategy [10] ($\alpha_X = 50/|X|$, $X = \{\mathcal{K}^{(c)}, \mathcal{K}^{(e)}, \mathcal{J}\}$, α_{w_c}=α_{w_e}=0.01). We set the number of content topics ($|\mathcal{K}^{(c)}|$) to 30 and number of expression topics ($|\mathcal{K}^{(e)}|$) to 10, the combination of which obtains the lowest perplexity value.

We investigate the topics learned by CETM. We check the top words in each content-topic & expression-topic and try to assign

Table 2: Top words in learned Expression & Content Topics. CETM extracts topics describing content & expression classes.

Content Topics			Expression Topics		
Class	**Topic**	**Top Words**	**Class**	**Topic**	**Top Words**
Affected Individuals	1	NUM, people, victims, families, friends, condolences	Tentativeness	4	report, appear, possible, most, perhaps, usually
	6	attack, lives, artist, NUM, killed, murder, countries		8	report, nearly, somehow, guess, dunno, apparent
Investigation	19	police, shooting, shot, gunman, suspect, video, confirm	Certainty	2	absolute, accurate, ain't, always, anytime, certain
	24	police, armed, funded, cops, photo, gas, forces, justice		5	ever, must, indeed, true, correct, absolute, obvious
Regions	26	place, authority, supermarket, blocked	Negation	4	not, don't, never, all, can't, nor, isn't, without
Event-Specific	11	pilot, cockpit, plane, door, fly, news, locked, airlines	Enquiring	1	what, why, how, when, not, where, which
	12	cafe, selfies, siege, hostages, scene, site, bystanders		7	why, what, how, not, when, likely, possible

them a content class and an expression class (Table 2). We observe 4 major content classes: **(a). Affected Individuals:** Information about killed/injured people, hostages. **(b). Investigations:** Police activities & investigations, status of criminals. **(c). Affected Regions:** Updates of the region where disaster has occurred and the status of its nearby locations. **(d). Other Event-Specific Updates:** Explicit updates of the incident, hospital numbers, etc. The major expression classes correspond to the four psycho-linguistic characteristics (**Tentativeness, Certainty, Negation,** & **Enquiring**). Table 2 contains the top words in a few sample topics with their respective classes. Note that *this work is one of the first in providing insights into the classes prevalent during man-made catastrophe* which are much different from natural disaster classes.

We derive a representation of each tweet in the latent space learned by CETM; the representation contains probability of that tweet belonging to the content-topics (describing content-classes) and the expression-topics (describing expression-classes). This representation is used as a feature in our tweet verification model (§4.3).

4.2 Incorporating Pre-Disaster User Behavior

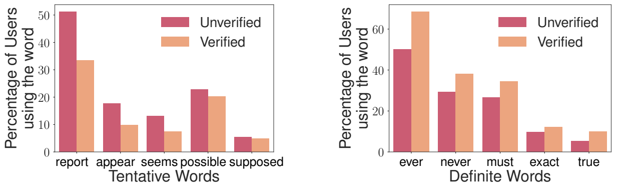
We next investigate user attributes relevant for tweet verification.

4.2.1 Static user attributes. Previous research works have mostly focused on certain Twitter-specific user attributes as an integral part of their tweet verification model [6, 52]. We check the distribution of four heavily used attributes (follower count, ratio of follower and following count, age of user i.e., time elapsed since the user joined Twitter, and status count i.e., total number of tweets posted) over users who posted verified and unverified tweets respectively (henceforth referred as verified and unverified users). We perform *Two-sample Kolmogorov-Smirnov test (ks2stat)* to check whether the difference between verified and unverified users is statistically significant. We obtained ks2stat score 0.044, 0.028, 0.046, and 0.090 (p-value > 0.1) for the above-mentioned four user features respectively. It is clearly evident that differences are not statistically significant and these static user features are unlikely to contribute much to the model efficiency, especially in the case of disasters.

4.2.2 Pre-Disaster behavior of users. We next inspect the behavior of users as observed on Twitter just before the time of disaster. For this, we first extract all the tweets posted by a user in the two months time range leading to the disaster using Twitter's Advanced Search functionality[1]. Similar to our tweet analysis, we check the degree of user's tentativeness & certainty in that time range based on the tweets extracted (Figure 2). We observe that a larger percentage of unverified users use tentative words before the disaster as compared against the verified class users. On the other hand, larger percentage of users who predominantly post verified tweets during disasters make use of definite words. The above analysis indicates that user's pre-disaster behavior is an important determining factor which can make her post unverified information during disasters and we incorporate this behavior in our model. We analyze the degree of user's psycho-linguistic characteristics (Tentativeness, Certainty, Negation, & Enquiring) by computing how frequently the user uses a word belonging to each of these four classes. We obtain a word list of these four classes using LIWC [33](same as §4.1.3). Let $\mathcal{W}_u^{(-i)}$ be the set of words in the tweets posted by user u, i days before the disaster. We define $r_u^{(c)}$, the regularity score of user u in class c, consisting of the set of words $\mathcal{W}^{(c)}$, as follows:

$$r_u^{(c)} = \left| \{ i \mid \mathcal{W}^{(c)} \cap \mathcal{W}_u^{(-i)} != \phi \text{ } and \text{ } 0 < i \leq 60 \} \right| \qquad (3)$$

i.e. the number of days user u has posted a word belonging to the class c in the 2 months time range. For each user u, we use regularity score of that user across the 4 psycho-linguistic classes ($r_u^{(c)}$) which accounts for user's pre-disaster behavior. These 4 scores act as 4 features which are used, along with representations in CETM's latent space, in our tweet stream verification model. Please note that the topic model (CETM) we developed in the last section cannot be used here as the tweets posted in the examined two months time range may not be disaster-based.



(a) Tentative Words

(b) Definite Words

Figure 2: Percentage of users (unverified/verified) using Tentative/Definite words in 2 month time range before the disaster. Unverified class users tend to use tentative words while verified class users use definite words.

4.2.3 Regularity score vis-a-vis verified-tweet-posting tendency. We examine if the regularity score, obtained using pre-disaster tweeting behavior of users, represents their verified and unverified tweet posting tendencies. We find the distributions of regularity scores of users who posted unverified and verified tweets respectively to be significantly distinct (two-sample KS test statistic values of 0.2539 and 0.2731 (p-value < 0.001) for tentativeness and certainty classes respectively).

We now describe our tweet stream verification model.

[1] https://twitter.com/search-advanced

4.3 Tweet verification using Tree-LSTMs

A source tweet which is to be verified, along with its set of replies forms a tree-like structure and while building the sequence model we aim to preserve this structure for effectively capturing the underlying nature of Twitter (generalizable to most social networks). For modeling tree-structured network topologies, Tai et al. [42] introduced *Tree-LSTM*, an extension to the basic LSTM architecture, where each LSTM unit incorporates information from multiple child units. An example of Tree-LSTM network corresponding to the tree-structure of replies is shown in Figure 3a. Here, each node is an LSTM unit which takes as input: (i) The representation of the tweet in the generated latent space + regularity scores of the user who posted it (x_i), and (ii) The hidden states of its child nodes. It uses them to update its own input gate, output gate, forget gate, hidden state and memory cell values.

In our tweet verification task, we work under the simplified assumption that only the source tweets (and not replies) are needed to be verified. And thus, we wish to derive probabilities of the source tweet s being verified and unverified. While working with Tree LSTMs, this would correspond to computing verification score for root node of each tree. At each root node s, we use a softmax classifier to obtain a probability distribution over verified and unverified classes, given the input $\{x\}_s$ observed in the tree with root node s. The classifier takes the hidden state h_s at the root node s as input and computes the probabilities as follows:

$$\hat{p_\theta}(y|\{x\}_s) = softmax(W^{(s)}h_s + b^{(s)}) \tag{4}$$

$$\hat{y_s} = \arg\max_y \hat{p_\theta}(y|\{x\}_s) \tag{5}$$

$$V(s) = \hat{p_\theta}(y = verified|\{x\}_s) \tag{6}$$

where $W^{(s)}$ and $b^{(s)}$ are the weight matrices and bias values respectively for the final tweet verification network, $\hat{y_s}$ is the predicted label (verified/unverified), & $V(s)$ is the verification score of tweet with root node s. The cost function is the negative log-likelihood of the true labels $y^{(s)}$ at each root node, defined for m training instances as:

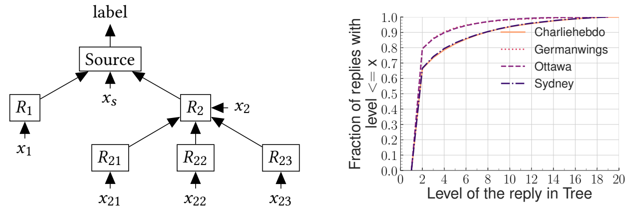
$$J(\theta) = -\frac{1}{m}\sum_{s=1}^{m} log\left(\hat{p_\theta}(y^{(s)}|\{x\}^s)\right) + \frac{\lambda}{2}||\theta||_2^2 \tag{7}$$

where λ is an L2 regularization hyperparameter. The height of the tree would affect the time complexities of training (back-propagating to the leaves) and testing (forward pass from leaves to root). We inspect the average height of the trees formed, described by the levels at which replies are present. Figure 3b shows the CDF of reply levels. As can be observed, 80-90% of the replies are at level $\leq$ 5 which acts as a limiting factor on complexities.

5 VERIFIED SUMMARIZATION OF TWEET STREAMS

We describe our disaster-specific verified tweet-stream summary generation system next.

Figure 3



(a) Example of Tree-LSTM Network with source tweet as root.

(b) CDF of Reply Levels. Most (80-90%) replies are at level ≤ 5

Table 3: Distribution of Situational/Non-Situational tweets across Verified/Unverified classes. Most (82%) of the unverified tweets are situational and thus, are part of summaries.

Disaster	Verified		Unverified	
	Situational	Non-Situational	Situational	Non-Situational
Charliehebdo	246	1494	392	92
Germanwings	91	141	217	21
Ottawa	113	311	393	93
Sydney	136	568	423	108
Total	586	2514	1425	314

5.1 Filtering Non-Situational Content

Prior works [34, 37] have shown that information posted on Twitter during disasters can be divided into two major classes — (i). **Situational** (information which provides updates about the current situation), and (ii). **Non-Situational** (sympathy or opinions of people). During disasters, end users like humanitarian organizations (OCHA, RedCross etc.) and government agencies are largely interested in situational updates and thus, in the context of crisis-specific summarization, summary of only tweets belonging to situational class are of primary importance.

We classify all the tweets in our dataset into situational and non-situational classes and remove the tweets belonging to the non-situational class. We use the situational tweet classifier developed by Rudra et al. [37] for classifying tweets[2]. Table 3 shows the statistics of situational and non-situational content present in the verified and unverified tweets[3]. Out of the 1739 unverified tweets, 1425 tweets (around 82%) are situational. This would mean that 82% of the unverified tweets would be a part of input streams provided to existing summarization frameworks and thus, might get inadvertently included in the summary. We try to minimize this unverified content by making use of tweet verification scores in our summarization framework, which we describe next.

5.2 Proposed Summarization Framework

The current state-of-the-art in real-time unsupervised disaster-specific extractive summarization of tweet streams is the Integer Linear Programming (ILP) based system proposed by Rudra et al [37] which tries to maximize the coverage of content words (nouns, numerals, main verbs, locations) in the summary (COWTS). A summary of L words is achieved by optimizing an ILP objective function, whereby the highest scoring tweets are returned as the output of summarization. Moreover, duplicate tweets are removed from the summarization framework and weight of the content words are multiplied by the binary indicators. This, in turn, brings diversity in the summary by capturing different content words. In this paper, we try to increase the proportion of verified tweets contained in their summary by utilizing the verification scores obtained through

[2] An SVM-based classifier utilizing a set of lexical and syntactic features.

[3] Ground-Truth labels of Verified & Unverified tweets have been used.

Tree-LSTMs (§4.3) in the objective function of COWTS. We multiply the indicator variable of each tweet with the verification score. The modified objective function for generating a summary of L words from n tweets having a total of m content words is:

$$max(\sum_{i=1}^{n} \boldsymbol{\gamma_v}.\mathbf{V(i)}.x_i \ + \ \sum_{j=1}^{m} Score(j).y_j) \tag{8}$$

subject to the constraints

$$\sum_{i=1}^{n} x_i \cdot Length(i) \leq L; \sum_{i \in T_j} x_i \geq y_j, \forall j; \sum_{j \in C_i} y_j \geq |C_i| \times x_i, \forall i \tag{9}$$

where x_i is the indicator variable of tweet i (1 if tweet i should be included in the summary, 0 otherwise), y_j is the indicator variable for content word j, $V(i)$ is the verification score of tweet i, $Score(j)$ is the tf-idf score of content word j normalized between 0 to 1, T_j is the set of tweets where content word j is present, and C_i is the set of content words present in tweet i. γ_v is a hyperparameter which controls the degree of verified content desired in the final summary. The objective function accounts for both, the likelihood of a tweet being verified (using $V(i)$) as well as the number of important content-words in the tweet (using $Score(j)$); γ_v allows the system to trade-off between these two factors. The three constraints ensure consistency w.r.t. desired length of summary and the inclusion or exclusion of tweets & content words. We term this new model **VERISUMM**. We use GUROBI Optimizer [14] to solve the ILP. After solving, the set of tweets i with $x_i = 1$, represent the summary.

5.3 Class-Regularized Verified Summaries

In §4.1.3, we discovered four content classes of tweets posted during man-made disasters — Affected Individuals, Investigations, Affected Regions, and Event-Specific. We may utilize the distribution of verified and unverified information over these content classes for improving the quality of the summaries. We devise a class-based *verification requirement regularizer* which takes into account the class-level insights in the summary generation process.

Let c_i denote the content-class i. Let $N_{c_i}^V$ & $N_{c_i}^U$ denote the number of verified & unverified tweets in the content-class i. Then, we compute the verification probability of a content class i, α_{c_i}, as:

$$\alpha_{c_i} = \frac{N_{c_i}^V}{N_{c_i}^V + N_{c_i}^U} \tag{10}$$

Next, we compute the *class-level verification requirement quotient*, β_{c_i}, as:

$$\beta_{c_i} = \frac{\min_j \alpha_{c_j}}{\alpha_{c_i}} \tag{11}$$

where min is taken over the 4 content-classes. The verification requirement quotient is inversely proportional to the amount of verified content in each content class; its value will be high for classes having tweets which are more prone to being unverified. We use this requirement quotient, β_{c_j}, as a regularizer to γ_v; the regularizer gives a data-driven control over γ_v. For the classes where number of verified tweets is high, the value of β_{c_i} will be low and thus, would decrease the value of γ_v. Similarly, for the classes where number of unverified tweets is higher, β_{c_i} will be high, thus increasing the value of γ_v (and consequently increasing the verified content proportion in the final summary). Using β_{c_i}, we modify the objective function of summary generation (Equation 8) as follows:

$$max(\sum_{i=1}^{n} \boldsymbol{\beta_{c_j}}.\gamma_v.V(i).x_i \ + \ \sum_{j=1}^{m} Score(j).y_j) \tag{12}$$

where c_j is the content class of tweet i. The constraints remain the same as Equation 9. We call this *class-regularized verified summarization system* — **VERISUMM++**. We will infer the α_{c_i} and β_{c_i} values for the four content classes in §7.2.

6 EVALUATING TWEET VERIFICATION MODULE

In this section, we evaluate the performance of our Tree-LSTM based tweet verification module. We also present certain statistics about different classes which we obtain using CETM.

6.1 Performance of Tree-LSTM on Verified Tweet Detection Task.

For training the Tree-LSTM model, we use a 128-dimensional single hidden layer at each LSTM unit, $learning\_rate = 0.05$, and $batch\_size = 50$. We train the model for 500 epochs. We compare our model against three state-of-the-art models for verified-tweet / fake-new detection: (i) **CRF**[52]: Employs Conditional Random Fields to learn from sequential tweet-user representations (word2vec for tweets & static twitter attributes for user), (ii) **RNN**[26]: Uses tweet clustering followed by RNN, tf.idf used for tweet representations, and (iii) **CSI**[35]: Integrates temporal patterns of tweets and representations of users computed using user's engagement graph. Moreover, we also evaluate the utility of different components of our model using the following variants: (i) **LDA-TL**: LDA [2] instead of CETM for tweet representations & Tree-LSTMs for modeling replies, (ii) **CETM-RNN**: tweet representations using CETM but employing a vanilla RNN for modeling replies, (iii) **CETM-TL**: tweet representations using CETM & Tree-LSTMs for modeling replies, (iv) **CETM-RS-TL**: user regularity score along with tweet representations using CETM as input features & Tree-LSTMs for modeling replies. Table 4 shows the accuracy and F1-score values obtained by the baselines and our models on the four events. Each of the dataset has been tested by training on only all the other datasets thus emulating the real-world scenario where supervision for the ongoing disaster is limited. CETM based variants perform better than all the baselines (5-13% in terms of accuracy and 3-13% in terms of F1-score); they also perform better than the fake-news detection systems (similar gains). RNN performs better than CSI as it also uses tweet clustering as part of its model.

LDA-TL performs significantly worse than all our remaining variants and a few baselines showing that Tree-LSTM alone cannot model the tweet verification task and needs rich tuned tweet representations as input. This is expected as we don't use a large dataset for training which hinders the automated learning of fruitful hidden representations. CETM-TL performs reasonably better than CETM-RNN indicating that modeling the inherent tree-like structure formed by replies is important. CETM-RS-TL performs the best in most cases which shows the utility of user's pre-disaster behavior in the tweet verification task.

Table 4: Performance of our tweet stream verification model & baselines. Rows highlighted in Yellow denote results from state-of-the-art fake news detection systems. Green row is our final model.

Model	Charliehebdo		Germanwings		Ottawa		Sydney	
	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score	Accuracy	F1-Score
CRF	0.715	0.570	0.678	0.696	0.718	0.721	0.717	0.640
RNN	0.694	0.621	0.617	0.642	0.711	0.729	0.704	0.639
CSI	0.758	0.605	0.502	0.612	0.515	0.649	0.594	0.625
LDA-TL	0.734	0.523	0.561	0.668	0.715	0.553	0.672	0.603
CETM-RNN	0.776	0.613	0.704	0.716	0.696	0.739	0.741	0.645
CETM-TL	0.787	0.656	**0.716**	0.721	**0.755**	**0.752**	0.737	0.686
CETM-RS-TL	**0.804**	**0.686**	0.702	**0.728**	0.745	0.740	**0.744**	**0.698**

Table 5: Variation in F1-Score over increasing deadlines. Performance reaches near saturation at deadline = 1hr.

Deadline	CRF	RNN	CSI	CETM-RNN	CETM-TL	CETM-RS-TL
T=0	0.656	0.658	0.615	0.58	0.58	0.62
T=1	0.656	0.658	0.621	0.676	0.68	0.70
T=2	0.656	0.658	0.63	0.665	0.681	0.691
T=3	0.656	0.658	0.625	0.671	0.699	0.711
T=4	0.656	0.658	0.623	0.678	0.702	0.718

6.2 Time Needed for Efficient Detection

Detection of unverified information at an early stage is very important, especially during disasters, so as to prevent its rapid propagation. As our system is reliant on replies, it might so happen that we need to wait for a significant amount of time for efficient detection. We find that, for most of the tweets in our dataset, 60% of the replies are posted within 1 hour and 90% are posted within 5 hrs of the source tweet. This is reasonable considering most disaster applications including summarization take snapshots of twitter stream at time intervals of one hour [37]. We reassert the same by testing our system after setting a deadline on the detection algorithm, where all the replies to the source tweet subsequent to the deadline are considered unavailable. Table 5 shows the average F1-score of our system variants with increasing deadlines. CRF and RNN don't have mechanism to handle the replies, hence have the same performance throughout. Performance is marginally low at deadline = 0 hrs, i.e., when no replies are available. CETM-RS-TL has magnified gains over other variants at deadline of 0 hrs as compared to gains at higher deadline values. This indicates that user's pre-disaster behavior is critical when other signals (such as replies) are missing. The performance values of all the variants reaches near saturation at the deadline of 1 hour; there are some local variations though which we attribute to noise in the replies.

6.3 Content-Class Identification: Distribution of Verified and Unverified Tweets

Using CETM, the tweets can be classified into the four content-classes; we use the following approach for identifying the content-classes. We manually mark each content-topic with a content-class (CETM identifies 30 content topics) and content class of a tweet is identified by its most probable content-topic. This classification of tweets into content-classes helps us in analyzing the distribution verified and unverified tweets have over these classes. Table 6 shows the class-wise distribution[4]. From Table 6, we make certain observations: **(a).** 34.9% (498 out of 1425) unverified tweets belong to the *Affected Individuals* content-class. However, only 15.5% (91 out of 586) verified tweets belong to the *Affected Individuals* class. **(b).** Around equal percentages of verified and unverified tweets belong to the *Investigations* (45.2% and 43.5% respectively) and *Affected Regions* (11.6% and 9.7% respectively) content-classes. **(c).** Around 31% (183 out of 586) of the verified tweets belong to the *Event-Specific* class. On the contrary, only 8% (116 out of 1425) of the unverified tweets are event-specific.

Table 6: Distribution of content-classes across verified (V) & unverified (Un-V) tweets. Last row shows the error %

Dataset	Individuals		Investigations		Regions		Ev.-Sp.	
	V	Un-V	V	Un-V	V	Un-V	V	Un-V
Charliehebdo	63	131	108	236	30	16	45	9
Germanwings	9	63	29	118	1	0	52	36
Ottawa	3	174	46	142	18	32	46	45
Sydney	16	130	72	149	8	118	40	26
Class Total	91	498	255	645	57	166	183	116
Class Error (%)	18.6%	14.8%	22.7%	19.8%	38.6%	36.7%	28.4%	43.1%

These observations indicate that the type of content helps in determining tweet verification likelihood. The observations are also used to generate high-quality summaries which we discuss in §7.2.

6.4 Analyzing the Errors

We study the errors committed by our verification module. We find some interesting patterns in how the error varies with content-classes (last row of Table 6). The error tendency is higher for *Affected Regions* & *Event-Specific* classes and is lower for *Affected Individuals* & *Investigations* classes. Most of the tweets belonging to the *Affected Individuals* report disaster statistics (# of victims) — low error rate in this class suggests that our model is robust to small variations in reported numbers (eg. 50 v/s 51 injured). A large number of speculations revolve around *Investigations* (43.5% in our dataset); a random sample indicates that speculations in *Investigations* receive more denials than other classes which helps our model in detecting them. The tweets belonging to *Affected Regions* & *Event-Specific* contain a lot of proper nouns (eg. hospital names) which might be one of the reasons for poor performance numbers; we will work on course-correction for these classes based on future data.

7 EVALUATING SUMMARIZATION SYSTEM

In this section, we describe the summarization baselines, evaluation metrics and discuss the performance of our proposed summarization module VERISUMM & its improved version VERISUMM++.

[4] We use the ground-truth labels of verified and unverified tweets. We only consider situational source tweets; non-situational source tweets & replies are not considered.

7.1 Performance of VERISUMM

7.1.1 Gold standard summaries. We create a gold-standard summary of 250 words for each event. We employ 3 volunteers working in the domain of disaster management. They individually prepare extractive summaries of the events. To generate the gold-standard summary from the 3 summaries, we first include tweets included in the summary of all the 3 volunteers, followed by the ones included by at least 2 until we achieve a summary of 250 words.

7.1.2 Quality of the summaries generated. We use the following three baselines for the summarization task: (i) **APSAL**[20]: Summarization using sentence salience prediction and affinity propagation [11] based clustering approach, (ii) **TSum4act**[31]: Summarization of actionable and informative tweets, (iii) **COWTS**[37]: ILP-based summarization maximizing content-words. All the three baselines are disaster-specific. For each event, we generate summaries of length 250 words using our models as well as the baselines. We evaluate the quality of the summaries generated based on the three criterion described below:

(1) Verified content proportion: We first compute the proportion of verified tweets in the summaries generated by VERISUMM (our model) at different γ_v values (1, 2, 5, & 10) (refer Eq.8) compare it against the 3 baselines. Table 7 shows the variation of verified proportion for all the datasets. VERISUMM consistently generates summaries which contain significantly more verified content than the baselines (12%-48% gain over best-performing baseline). The verified content proportion increases on using higher γ_v values.

(2) ROUGE-1 w.r.t. Ground-Truth summaries: We use the standard ROUGE [24] metric to measure the overlap of summaries generated by respective models with the ground-truth summaries (both 250 words). Due to the informal nature of tweets, we measure F-score of only ROUGE-1 variant. We compare the scores obtained by VERISUMM (for different γ_v values) with the baselines. Table 7 contains the ROUGE-1 F-scores of the different models. We obtain significant gains over best-performing baselines (7.6% - 13.5%). The gains decrease for higher values of γ_v with $\gamma_v = 10$ performing slightly worse than most baselines.

(3) Richness of the summaries: Finally, we check if the verified framework has any effect on the richness of the generated summary. We compute richness as the ratio of number of content words and the total number of words in the summary (where content words are as defined in §4.1). Table 7 contains the richness values for variants of VERISUMM as well as the baselines. We achieve richness values at par with baselines (-1.9% to 6.3% gain).

7.1.3 Discussion on the results. The above results clearly indicate that VERISUMM outperforms the current state-of-the-art disaster-specific summarization systems in terms of generating verified content. Furthermore, at the same time, it is able to maintain high scores on the other important quality measures (richness & ROUGE score). The hyperparamater γ_v acts as a trade-off between these three quality metrics (high $\gamma_v \implies$ higher verified content but lower richness & ROUGE scores). We observe that a balance between these three metrics can be maintained by choosing γ_v close to 5 at which VERISUMM generates summaries which are highly verified and superior to the state-of-the-art in terms of richness & ROUGE scores. In §7.2, we take insights from the discovered content-classes in order to improve ROUGE-1 scores & richness while maintaining similar verified proportions.

7.2 Improvements using Class-Level Insights: Evaluating VERISUMM++

We now evaluate the effect of incorporating class-level insights, as described in §5.3, on the summaries. We infer the α_{c_i} and β_{c_i} values for the four content classes using the distribution of verified and unverified content presented in Table 6. The α_{c_i} values for *Affected Individuals*, *Investigations*, *Affected Regions* and *Event-Specific* are 0.154, 0.283, 0.255, and 0.612 respectively. The corresponding β_{c_i} values are 1, 0.544, 0.604, and 0.252. We present the result of VERISUMM (5)++ (i.e. γ_v as 5 - similar trend for other values of γ_v) in the sixth row (just below VERISUMM (5)) in Table 7. It shows the verified proportion, ROUGE-1, and richness values. VERISUMM++ in almost all the cases improves the verified proportion but more importantly helps us in attaining improved ROUGE-1 scores (0%-8% gain) and richness values (0%-3% gain).

7.3 Reassessing Unverified Information in Summaries

Table 1 reports the proportion of verified tweets with respect to the gold standard dataset - the dataset however is limited by the time of its generation; the unverified tweets might have become verified over time. Hence, we *manually* tried to relabel the unverified tweets. One way [28] of figuring out the current labels is by using rumor debunking sites such as scopes.com. However, their coverage for disaster-related tweet is not high, especially the ones used in our study. Hence, we explore a list of 10 news sources (BBC, New York Times, CNN, Guardian, The Washington Post, ABC News, Sky News, Fox News, 9News for Sydney, & CBC for Ottawa) to label the unverified tweets. We collect articles, related to the unverified information, posted by any one of the 10 news sources. If at least one article confirms the information by either reporting first-hand experiences of users or by verifying it after having reported it as unconfirmed, we change the label of the tweet to verified. Note that small variations in reported numbers are ignored (eg. 11 casualties v/s 12 casualties). However, if any of the articles refute the posted information, we don't change the label. Also, if all the articles tentatively put their claims regarding the information ('unconfirmed', 'not verified', 'as per unknown sources'), the label is not changed.

After getting the *eventual labels* of the unverified tweets in the summaries, we can compute the **eventual verified proportions**. The eventual verified proportion of a summary is given by the number of tweets in the summary which were verified (by gold standard) + tweets which were initially unverified but got re-labeled as verified divided by the total number of tweets in the summary. Table 8 reports the values of the eventual verified proportions of our model variants and baselines. We perform 5% - 21% better than the baselines in eventual verified proportion. More significantly, we find that VERISUMM (1) outweighs the closest baseline (COWTS) in the proportion of eventually verified articles although it was behind when result was obtained only on gold standard data (check third and fourth rows of Table 7). This implies that even though a significant proportion of summary tweets generated by some variants of VERISUMM (1) may be unverified as per initial analysis (done during or just after the disaster), the probability of those

Table 7: Quality of Summaries generated (in terms of Proportion of Verified Tweets, ROUGE-1 F-score, and Richness) by VERISUMM (γ_v) for different γ_v values, VERISUMM++ and three baselines (APSAL, TSum4act, and COWTS). Row highlighted in Green shows results for VERISUMM++.

Model	Charliehebdo			Germanwings			Ottawa			Sydney		
	VerProp	ROUGE-1	Richness	VerProp	ROUGE-1	Richness	VerProp	ROUGE-1	Richness	VerProp	ROUGE-1	Richness
APSAL	0.85	0.329	0.503	0.444	0.44	0.577	0.353	0.309	0.576	0.412	0.341	0.445
TSum4act	0.714	0.312	0.458	0.556	0.450	0.533	0.210	0.292	0.595	0.588	0.380	0.456
COWTS	0.75	0.351	**0.680**	0.389	0.444	0.637	0.40	0.341	0.619	0.56	0.365	0.635
VERISUMM (1)	0.714	0.362	0.667	0.412	0.4621	**0.654**	0.411	**0.387**	0.611	0.583	**0.431**	0.6311
VERISUMM (2)	0.837	0.368	0.667	0.5	0.511	0.645	0.428	0.354	**0.658**	0.64	0.409	**0.642**
VERISUMM (5)	0.930	0.378	0.658	0.588	0.488	0.624	0.56	0.346	0.624	0.769	0.372	0.622
VERISUMM (5) ++	0.933	**0.382**	0.662	0.611	**0.529**	0.64	0.576	0.352	0.638	0.731	0.374	0.622
VERISUMM (10)	**0.956**	0.332	0.573	**0.823**	0.450	0.533	**0.642**	0.292	0.572	**0.786**	0.311	0.585

Figure 4: Comparison of Summary Tweets of Ottawa retrieved by VERISUMM++ and COWTS along with their verification scores; 4 representative tweets are shown here for both. Tweets with verification score in blue are verified, red are unverified and brown are eventually verified (after re-labeling). Note the use of tentative words (reports, claiming) in unverified tweets.

VERISUMM (5) ++	COWTS
T1: The RCMP intervention team members Parliament Hill <link>. - **0.824**	**T1:** The RCMP intervention team members Parliament Hill <link>. - **0.824**
T2: Canadian Prime Minister Stephen Harper is due to make a statement shortly, **reports** say - **0.375**	**T2:** Canadian Prime Minister Stephen Harper is due to make a statement shortly, **reports** say - **0.375**
T3: Canadian police say #OttawaShooting "caught us by surprise" - **0.772**	**T3:** ISIS Media account posts picture **claiming** to be Michael Zehaf-Bibeau, dead #OttawaShooting suspect. - **0.174**
T4: Watch video showing gunfire inside Canada's parliament in Ottawa <link> - **0.639**	**T4:** Sergeant-at-Arms Kevin Vickers hailed as hero for shooting Canadian Parliament gunman: <link> - **0.311**

Table 8: Eventual Verified Proportions of Summaries generated by VERISUMM (γ_v), VERISUMM (5) ++ and baselines.

System	Charliehebdo	Germanwings	Ottawa	Sydney
APSAL	0.9	0.722	0.765	0.765
TSum4act	0.928	0.833	0.684	0.823
COWTS	0.875	0.722	0.8	0.76
VERISUMM (1)	0.893	0.823	0.791	0.875
VERISUMM (2)	0.946	0.833	0.905	0.88
VERISUMM (5)	**0.977**	**0.941**	0.96	0.923
VERISUMM (5) ++	**0.977**	**0.944**	0.962	0.923
VERISUMM (10)	**0.977**	**0.941**	**0.964**	**0.928**

tweets being ultimately found authentic is higher for VERISUMM than for COWTS. Moreover, the proportions for $\gamma_v = 5$ and $\gamma_v = 10$ are statistically indistinguishable. This reiterates the fact that summarization model performs the best for value of γ_v close to 5.

7.4 Representative Summaries

We summarize the difference in output of our final system — **VERISUMM++** and the most competitive baseline COWTS through illustration of (part of) the summaries produced by them. In general, VERISUMM++ captures tweets having high verification scores compared to COWTS. Specifically, we observe the following four patterns in these summaries as highlighted in Figure 4 — **(T1)**. a number of verified tweets are captured by both VERISUMM++ and COWTS, **(T2)**. some unverified tweets are retrieved by both systems, **(T3)**. there are unverified tweets present in the summary of COWTS which are not shortlisted in VERISUMM++; they are replaced by suitable verified tweets in VERISUMM++. **(T4)**. some tweets in summaries initially stay unverified but in VERISUMM++ eventually get verified (when re-annotated as per §7.3); most of the unverified tweets continue to stay unverified in COWTS.

Case Study of the 2019 Sri Lankan Easter Attacks: To further analyze the robustness of our verification-summarization framework, we present an interesting case study of the recent Sri Lankan Attacks. Immediately following the attacks, a Sri Lankan minister tweeted that a foreign intelligence report predicting the attacks was noted to some officials few days before the attack[5]. Some people on Twitter questioned the authenticity of this tweet while others started speculating the names of Sri Lankan officials who were aware/unaware of this report; the names included President & Prime Minister. Both the original tweet by the minister and the subsequent speculative tweets were initially unverified as there was no supporting data to authenticate them. Two days later, both the President & the PM denied being informed about the report but verified that the report was known to few security officers[6]. This meant that the basic content of minister's tweet was true (unverified initially, eventually true) but most of the subsequent tweets were false — The ideal summary would not include these tweets. We generate and analyze summaries of the 2019 Sri Lankan Attacks using VERISUMM++ & COWTS. The unverified information related to the speculations around intelligence reports is part of COWTS's summary but not that of VERISUMM++.

8 CONCLUSION & FUTURE CHALLENGES

To the best of our knowledge, this is the first work on generating verified summaries of tweet streams during disasters. The simple

[5] https://bit.ly/2V7xH8m

[6] https://bbc.in/2JEVNjS; https://bit.ly/2W2DICY

but novel content-expression topic model (CETM) which simultaneously incorporates tweet's content and its way of expression for creating tweet representations is at the core of the innovation. In the process, we discovered four content classes of information posted during man-made disasters. The tweet representations and pre-disaster user behavior (regularity scores) were used to train a Tree-based LSTM model, with an objective of inferring tweet verification probabilities. The verification scores and the information content and the class information of the tweets were used in an ILP framework for generating the desired verified summaries. As expected, our summaries contained exceptionally high proportion of verified information; but more interestingly the summaries also had better ROUGE-1 scores and richness than the state-of-the-art. Also, the proportion of eventually verified tweets included in the summary (which cannot be verified at the time of usage) is much higher than competing techniques. We believe the technique developed in this paper has wider implication and usage. The technique can potentially be used during natural disasters, epidemics; can be personalized according to stakeholders requirement - we would explore those possibilities as one of our immediate future works.

REFERENCES

[1] A. Arif, JJ. Robinson, SA. Stanek, ES. Fichet, P. Townsend, Z. Worku, and K. Starbird. 2017. A Closer Look at the Self-Correcting Crowd: Examining Corrections in Online Rumors. In *Proc. CSCW*. 155–168.

[2] DM. Blei, AY. Ng, and MI. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3 (2003), 993–1022.

[3] K. Bontcheva, L. Derczynski, A. Funk, MA. Greenwood, D. Maynard, and N. Aswani. 2013. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *Proc. RANLP*.

[4] MA. Cameron, R. Power, B. Robinson, and J. Yin. 2012. Emergency situation awareness from twitter for crisis management. In *Proc. WWW*. 695–698.

[5] C. Castillo. 2016. *Big Crisis Data: Social Media in Disasters and Time-Critical Situations.* Cambridge University Press.

[6] C. Castillo, M. Mendoza, and B. Poblete. 2011. Information credibility on twitter. In *Proc. WWW*. 675–684.

[7] D. Chakrabarti and K. Punera. 2011. Event summarization using tweets. In *Proc. ICWSM*.

[8] T. Chen, L. Wu, X. Li, J. Zhang, H. Yin, and Y. Wang. 2017. Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection. *arXiv preprint arXiv:1704.05973* (2017).

[9] M. Conover, J. Ratkiewicz, MR. Francisco, B. Gonçalves, F. Menczer, and A. Flammini. 2011. Political polarization on twitter. *ICWSM* 133 (2011), 89–96.

[10] Q. Diao, J. Jiang, F. Zhu, and EP. Lim. 2012. Finding bursty topics from microblogs. In *Proc. ACL*. 536–544.

[11] BJ. Frey and D. Dueck. 2007. Clustering by passing messages between data points. *science* 315, 5814 (2007), 972–976.

[12] A. Friggeri, L. Adamic, D. Eckles, and J. Cheng. 2014. Rumor Cascades.. In *Proc. ICWSM*.

[13] A. Gupta, H. Lamba, P. Kumaraguru, and A. Joshi. 2013. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proc. WWW*. 729–736.

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

[15] A. Hannak, D. Margolin, B. Keegan, and I. Weber. 2014. Get Back! You Don't Know Me Like That: The Social Mediation of Fact Checking Interventions in Twitter Conversations.. In *ICWSM*.

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

[17] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg. 2014. AIDR: Artificial intelligence for disaster response. In *Proc. WWW*. 159–162.

[18] M. Imran, P. Mitra, and C. Castillo. 2016. Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. In *Proc. LREC*.

[19] M. Jang and J. Allan. 2018. Explaining Controversy on Social Media via Stance Summarization. In *Proc. SIGIR*. 1221–1224.

[20] C. Kedzie, K. McKeown, and F. Diaz. 2015. Predicting salient updates for disaster summarization. In *Proc. ACL*, Vol. 1. 1608–1617.

[21] MAH. Khan, D. Bollegala, G. Liu, and K. Sezaki. 2013. Multi-Tweet Summarization of Real-Time Events. In *Proc. IEEE Socialcom*. 128–133.

[22] A. Kohut, DC. Doherty, M. Dimock, et al. 2009 (Accessed 2018). *PRESS ACCURACY RATING HITS TWO DECADE LOW.*

[23] S. Kwon, M. Cha, K. Jung, W. Chen, and Y. Wang. 2013. Prominent features of rumor propagation in online social media. In *Proc. ICDM*. 1103–1108.

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

[25] X. Liu, A. Nourbakhsh, Q. Li, R. Fang, and S. Shah. 2015. Real-time rumor debunking on twitter. In *Proc. CIKM*. 1867–1870.

[26] J. Ma, W. Gao, P. Mitra, S. Kwon, BJ. Jansen, KF. Wong, and M. Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks.. In *IJCAI*. 3818–3824.

[27] J. Ma, W. Gao, Z. Wei, Y. Lu, and KF. Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proc. CIKM*. 1751–1754.

[28] J. Ma, W. Gao, and KF. Wong. 2017. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *Proc. ACL*. 708–717.

[29] M. Mendoza, B. Poblete, and C. Castillo. 2010. Twitter Under Crisis: Can we trust what we RT?. In *Proceedings of the first workshop on social media analytics*. 71–79.

[30] G. Neubig, Y. Matsubayashi, M. Hagiwara, and K. Murakami. 2011. Safety Information Mining-What can NLP do in a disaster-.. In *IJCNLP*, Vol. 11. 965–973.

[31] MT. Nguyen, A. Kitamoto, and TT. Nguyen. 2015. Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. In *Proc. PAKDD*. 64–75.

[32] MT. Nguyen, A. Kitamoto, and TT. Nguyen. 2015. TSum4act: A Framework for Retrieving and Summarizing Actionable Tweets During a Disaster for Reaction. In *Proc. PAKDD*. 64–75.

[33] JW. Pennebaker, RJ. Booth, and ME. Francis. 2007. LIWC2007: Linguistic inquiry and word count. *Austin, Texas: liwc. net* (2007).

[34] Y. Qu, C. Huang, P. Zhang, and J. Zhang. 2011. Microblogging after a major disaster in China: a case study of the 2010 Yushu earthquake. In *Proc. CSCW*. 25–34.

[35] N. Ruchansky, S. Seo, and Y. Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In *Proc. CIKM*. 797–806.

[36] K. Rudra, S. Banerjee, N. Ganguly, P. Goyal, M. Imran, and P. Mitra. 2016. Summarizing Situational Tweets in Crisis Scenario. In *Proc. HT*. 137–147.

[37] K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh. 2015. Extracting situational information from microblogs during disaster events: a classification-summarization approach. In *Proc. CIKM*. 583–592.

[38] K. Rudra, P. Goyal, N. Ganguly, P. Mitra, and M. Imran. 2018. Identifying Sub-events and Summarizing Disaster-Related Information from Microblogs. In *Proc. SIGIR*. 265–274.

[39] O. Shapira, H. Ronen, M. Adler, Y. Amsterdamer, J. Bar-Ilan, and title = Dagan, I.âĂİ. [n. d.].

[40] K. Starbird, D. Dailey, O. Mohamed, G. Lee, and E. Spiro. 2018. Engage Early, Correct More: How Journalists Participate in False Rumors Online During Crisis Events. In *Proc. CHI*. 105:1–105:12.

[41] S. Sun, H. Liu, J. He, and X. Du. 2013. Detecting event rumors on sina weibo automatically. In *Asia-Pacific Web Conference*. 120–131.

[42] KS. Tai, R. Socher, and CD. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. ACL*.

[43] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. 2010. Microblogging During Two Natural Hazards Events: What Twitter May Contribute to Situational Awareness. In *Proc. SIGCHI*. 1079–1088.

[44] A. Vlachos and S. Riedel. 2015. Identification and verification of simple claims about statistical properties. In *Proc. EMNLP*. 2596–2601.

[45] Nguyen Vo and Kyumin Lee. 2018. The Rise of Guardians: Fact-checking URL Recommendation to Combat Fake News. In *The 41st International ACM SIGIR Conference on Research &#38; Development in Information Retrieval*. 275–284.

[46] L. Wu and H. Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proc. WSDM*. 637–645.

[47] Y. Wu, PK. Agarwal, C. Li, J. Yang, and C. Yu. 2014. Toward computational fact-checking. *Proceedings of the VLDB Endowment* 7, 7 (2014), 589–600.

[48] F. Yang, Y. Liu, X. Yu, and M. Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*. 13.

[49] L. Zeng, K. Starbird, and E. Spiro. 2016. Rumors at the speed of light? Modeling the rate of rumor transmission during crisis. In *System Sciences (Proc. HICSSS*. 1969–1978.

[50] L. Zeng, K. Starbird, and E. Spiro. 2016. #Unconfirmed: Classifying Rumor Stance in Crisis-Related Social Media Messages. In *Proc. ICWSM*. 747–750.

[51] Z. Zhao, P. Resnick, and Q. Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proc. WWW*. 1395–1405.

[52] A. Zubiaga, M. Liakata, and R. Procter. 2017. Exploiting Context for Rumour Detection in Social Media. In *International Conference on Social Informatics*. 109–123.

[53] A. Zubiaga, M. Liakata, R. Procter, K. Bontcheva, and P. Tolmie. 2015. Crowdsourcing the annotation of rumourous conversations in social media. In *Proc. WWW*. 347–353.