**ABSTRACT**

Occurrences of catastrophes such as natural or man-made disasters trigger the spread of rumours over social media at a rapid pace. Presenting a trustworthy and summarized account of the unfolding event in near real-time to the consumers of such potentially unreliable information thus becomes an important task. In this work, we propose MTLTS, the first end-to-end solution for the task that jointly determines the credibility and summary-worthiness of tweets. Our credibility verifier is designed to recursively learn the structural properties of a Twitter conversation cascade, along with the stances of replies towards the source tweet. We then take a hierarchical multi-task learning approach, where the verifier is trained at a lower layer, and the summarizer is trained at a deeper layer where it utilizes the verifier predictions to determine the salience of a tweet. Different from existing disaster-specific summarizers, we model tweet summarization as a supervised task. Such an approach can automatically learn summary-worthy features, and can therefore generalize well across domains. When trained on the PHEME dataset [29], not only do we outperform the strongest baselines for the auxiliary task of verification/rumour detection, we also achieve 21 - 35% gains in the verified ratio of summary tweets, and 16 - 20% gains in ROUGE1-F1 scores over the existing state-of-the-art solutions for the primary task of trustworthy summarization.

**CCS CONCEPTS**

- Computing methodologies → Natural language processing; Information systems → Summarization.

Permission to make hard or digital copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

**KEYWORDS**

Trustworthy Summarization, Rumour Detection, Disaster, Twitter

ACM Reference Format:

Rajdeep Mukherjee, Uppada Vishnu, Hari Chandana Peruri, Sourangshu Bhattacharya, Koustant Rudra, Pawan Goyal, and Niloy Ganguly. 2022. MTLTS: A Multi-Task Framework To Obtain Trustworthy Summaries From Crisis-Related Microblogs. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM ’22), February 21–25, 2022, Tempe, AZ, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498536

1 INTRODUCTION

Sudden adversities such as natural or man-made disasters escalate the speed of news sharing over social media in the form of posts containing critical situational updates, information about victims, and general public sentiments, among others [6]. Such platforms are however prone to widespread proliferation of misinformation, given their unmoderated nature, and the unavailability of sufficient updates from reliable sources to verify the credibility of messages at the time of posting [27]. In such time-critical scenarios, it is nearly impossible for any concerned stakeholder, including humanitarian organizations, crisis-responders, and public in general, to manually fact-check such overwhelming volume of information [15]. Motivated by these factors, and acknowledging the far-reaching detrimental consequences of falsehood [11], we propose MTLTS, an end-to-end solution that jointly verifies and summarizes large volumes of disaster-related tweets to generate information-rich and trustworthy summaries in near real-time.

Existing methods [7, 21, 23, 24] to summarize disaster-specific microblogs do not make any explicit efforts to validate the authenticity of summary tweets, thereby making their solutions unreliable. To the best of our knowledge, VERISUMM [25] is the only proper baseline for the task of verified/trustworthy summarization. However, it takes a pipeline approach, and hence suffers from error propagation between the steps. We on the other hand jointly determine the credibility and summary-worthiness of a tweet by training MTLTS using a hierarchical multi-task learning setup. Such an approach intuitively mimics a pipeline while addressing its limitations. Prior disaster-specific summarizers (including VERISUMM)
are additionally unsupervised in nature where the summaries are built by maximizing the presence of event-specific keywords. It however affects their ability to generalize across unseen events, presumably with their own set of information classes and associated keywords. To overcome this shortcoming, we design a supervised approach, as it can implicitly learn summary-worthy features along with the signals to identify a potentially rumourous tweet when trained jointly with the verification subtask.

Compared to the extensive literature on text summarization [5], methods focusing on summarizing social media texts are limited [12]. In this work, we additionally make an attempt to address this gap by leveraging SummaRuNNer [18], one of the seminal works on extractive document summarization, and suitably modifying it to suit our purpose. More specifically, we need document-summary pairs to train the model, where documents consist of multiple sentences (tweets in our case). Here, instead of considering each tweet as a separate document or the entire tweet stream as a single document, we randomly select a subset of tweets to define a pseudo document. However, by doing so, we restrict a tweet’s context to the document it is part of. We therefore allow the same tweet to be included in multiple such documents as long as each tweet is included in at least m documents, where m is a hyper-parameter. Finally, we propose a novel approach to approximate a tweet’s global context from its multiple document-level local contexts.

Our end goal is to generate a summary with a high ratio of verified or non-rumourous tweets to total tweets. The efficacy of our credibility verifier is therefore of prime importance as it directly impacts the performance of the joint task. Twitter conversation threads possess a self-correcting nature as the true veracity of a claim gets revealed gradually through collective sense-making [27]. Many prior works have tried to exploit this phenomenon [10, 13, 15, 25] by using tree-structured neural networks such as Tree-LSTMs [26] to recursively capture the structural characteristics of such an information cascade. Ducci et. al. [3] recently proposed Cascade-LSTM, a novel bi-directional variant of Tree-LSTM, to learn the complete cascade. We extend it to additionally detect the stances of each of the reply nodes towards the potentially rumourous source tweet [28], to complete the design of our verifier module. The social signals learnt, and the predictions made by the verifier (trained at a lower layer) are used by the summarizer (trained at a deeper layer) to determine the summary-worthiness of a tweet. Tweet-level semantic features are learnt using BERTweet [19], whose parameters are jointly updated by our multi-task objective. We use ILP to obtain the final summary by maximizing the relevance scores of tweets thus obtained, while minimizing redundancy.

Our overall contributions can be summarized as follows:

- We present MTLTS, the first end-to-end solution to obtain trustworthy summaries from large volumes of disaster-related tweets. Using multi-task learning, MTLTS is trained to jointly determine the credibility and summary-worthiness of a tweet.
- Our approach is supervised, which enhances the generalizability of our solution to unseen events. More importantly, we present a novel way to leverage an extractive document summarization technique for summarizing event (here, disaster)-specific tweets.
- Our credibility verifier improves upon the current state-of-the-art for the rumour detection task by jointly learning the structural characteristics of information cascades, and stances of replies towards the source tweet. The features thus learnt help the verifier and the summarizer alike in respectively determining the veracity and summary-worthiness of source tweets.
- Extensive experiments on PHEME [29] establish the better efficacy of MTLTS over strong state-of-the-art baselines. Comparative results along with the favourable outcomes of a qualitative human evaluation experiment are discussed in Section 4.

We present the design of MTLTS in Section 2. Our experimental setup, additional related work, and baselines are discussed in Section 3. While our proposed solution is generic, we perform a thorough analysis of MTLTS-generated summaries across several disaster-specific classes relevant for PHEME events in Section 5.3, and discover that the verified ratio of summaries generated from Police Investigation-related tweets substantially improves over time as authoritative sources gradually confirm or deny initially unverified claims. We conclude our work in Section 5.4 with a case study.

2 METHODOLOGY

2.1 Problem Formulation

Let \( T = \{T_1, T_2, ..., T_T\} \) represent a dataset of Twitter conversation threads posted during a crisis event, with \( T \) representing the length of the dataset. Here, each thread \( T_i = \{s_i, r_{i1}, r_{i2}, ..., r_{in}\} \) consists of a source tweet or claim \( s_i \) whose credibility needs to be verified. Further, we need to decide if \( s_i \) should be included in the summary. The reply tweets \( r_{is} \) are direct or indirect responses to \( s_i \), thereby collectively forming a propagation tree like structure with \( s_i \) as their root. Our multi-task objective is formulated as a supervised classification problem, which learns a classifier \( f \) from labeled source tweets, \( f : s_i \rightarrow \{\text{verified, unverified}\} \) and \( \text{Summ}_i \in \{\text{in-summary, out-of-summary}\} \). We also capture the confidence scores \( \text{Ver}_i^{\text{prob}} \) and \( \text{Summ}_i^{\text{prob}} \) with which the final labels were predicted by the respective modules. These scores are used in the inference phase to generate the final summary.

2.2 MTLTS Architecture

Fig. 1 illustrates the building blocks of our architecture, with Fig. 1(c) showing the way they are connected in MTLTS. We now detail below our model components and methodology for the joint task.

2.2.1 Shared Text Encoder. We use pre-trained BERTweet [19] as our text encoder. A given tweet is first tokenized by the BERTweet-Tokenizer, prepended with the special [CLS] token, and passed through the encoder, which then learns rich contextualized representations for the tweet’s constituent tokens. Given that both our modules are supervised classification tasks, we consider the 768-dimensional embedding of the [CLS] token from the last layer of BERTweet as the final semantic representation of the tweet. BERTweet-parameters are respectively updated by the two modules when trained separately. For training MTLTS, the text encoder is shared between the verifier and the summarizer, and the parameters are jointly fine-tuned by our multi-task objective, thereby facilitating interaction between the two modules.

2.2.2 Tweet Verification. Given a Twitter conversation thread \( T_i \), the primary task of the credibility verifier is to classify the source tweet \( s_i \) as either verified/non-rumourous or unverified/rumourous.
The secondary task is to classify each of the response tweets as either supporting, denying, querying, or commenting. For this task, we use cross-entropy to define the loss function, $L_{\text{stance}}$. For the verification/rumour detection task, the concatenated feature vector $[h_{i1} : h_{i\text{leaf}}] : [CLS]_{i\text{th}}$ is passed through a feed-forward layer with softmax at the root node of the tree to obtain the probability distributions for $s_i$ over the two classes, verified and unverified.

Here, $h_{i\text{leaf}}$ represents the average over the top-down hidden states of all leaves. Binary Cross-Entropy is used to define the loss function, $L_{\text{rumour}}$. The verifier is trained by optimizing the joint loss function, $L_{\text{VER}}$, with $\lambda_1$ being the trade-off parameter between the two losses.

$$L_{\text{VER}} = L_{\text{rumour}} + \lambda_1 \cdot L_{\text{stance}}$$

### 2.2.3 Tweet Summarization

We model tweet summarization as a supervised classification task by leveraging SummaRuNNer [18]. It consists of a two-layer bi-directional GRU-RNN. The first layer works at the word-level to learn contextualized word representations which are then average-pooled to obtain sentence representations. We replace this layer by our BERTTweet-based text encoder, that is fine-tuned to obtain tweet semantic representations. The second layer of bi-directional GRU-RNN works at the sentence-level (tweet-level in our case) to learn the contextualized representations of tweets in a document. Graphical representation of our modified network is depicted under Tweet Summarizer module in Fig. 1.

We now define the notion of a document. Let $S = \{s_1, s_2, \ldots, s_T\}$ represent the set of all source tweets present in the set of tweet threads, $T$ with $|T|$ representing the dataset length. Our goal is to produce an extractive summary from $S$ while maximizing the verified ratio, content-richness, and novelty of selected tweets, and minimizing redundancy. To efficiently train the summarizer, we create multiple documents, each consisting of a subset of available tweets. However by doing so, we restrict a tweet’s context to the document it is part of. We explain the working of our model with this setup, before presenting a novel method to approximate the global context of a tweet $s_i$ from its document-level contexts. Let $D_k = \{s_1, s_2, \ldots, s_{DL}\}$ represent the $k^{th}$ document with length $DL$. All tweets in $D_k$ are sorted in ascending order of their posting time to learn any possible temporal clues. Following [18], we obtain the document representation $d_k$ using the hidden states of the sentence-level bi-directional GRU-RNN as follows:

$$d_k = \text{tanh}(W_d \frac{1}{DL} \sum_{i=1}^{DL} h_k^i + b)$$

Here, $h_k^i$ represents the hidden state vector corresponding to $s_i \in D_k$. $W_d$ and $b$ respectively represent the weight and bias parameters. A classification layer (Fig. 1) then takes a binary decision regarding the summary-worthiness of $s_i$ by sequentially revisiting the tweets in a second pass. Probability ($p_{DL}^i$) of $s_i$ to be included in the summary for document $D_k$ is obtained as:

$$p_{DL}^i = P(y_{DL}^i = 1 | h_k^i, \text{sum}_k^i, Ver_{i\text{prob}})$$

The value of binary variable $y_{DL}^i$ determines the inclusion/exclusion of tweet $s_i$ in the summary for $D_k$. Here, $\text{sum}_k^i$ represents the intermediate representation of the summary formed till $s_i$ is visited:

$$\text{sum}_k^i = \sum_{j=1}^{i-1} h_k^j p_{DL}^j$$

Therefore, for a given tweet \( s_i \in D_k \), its probability of being verified \((V_{i}^{\text{prob}})\) as predicted by the underlying verifier, its content \((h_k^i)\), its importance given the context of the document \((d_k)\), and possibility of adding redundant information considering the summary already formed \((\text{sum}_k^i)\) are all taken into account while deciding upon its summary membership. We request our readers to refer to [18] for further details about the equations.

\[ p_i^D_k \text{ thus obtained is however limited to the context of } D_k. \]

In the second step, these scores are used (Eq. 3), while training the summarizer to co-train the two modules in a hierarchical multi-task learning setup:

\[ L_{\text{MTLTS}} = L_{\text{VER}} + \lambda_2 \cdot L_{\text{SUMM}} \]

Here, \( \lambda_2 \) represents a trade-off between the two losses. The verifier is trained in batches of tweets, whereas the summarizer is trained in batches of documents. Each training epoch is therefore divided into two steps. First, we train the verifier at a lower layer to obtain the probability score \((V_{i}^{\text{prob}})\) of source tweet \( s_i \) being non-rumourous. In the second step, these scores are used (Eq. 3), while training the summarizer at a deeper layer, to decide the summary-worthiness of tweets. As the joint loss backpropagates through the shared encoder, BERTweet parameters are fine-tuned end-to-end to produce latent representations of tweets which benefit both the tasks.

\subsection{Joint Training}

We minimize the following joint loss to co-train the two modules in a hierarchical multi-task learning setup:

\[ L_{\text{MTLTS}} = L_{\text{VER}} + \lambda_2 \cdot L_{\text{SUMM}} \]

Here, \( \lambda_2 \) represents a trade-off between the two losses. The verifier is trained in batches of tweets, whereas the summarizer is trained in batches of documents. Each training epoch is therefore divided into two steps. First, we train the verifier at a lower layer to obtain the probability score \((V_{i}^{\text{prob}})\) of source tweet \( s_i \) being non-rumourous. In the second step, these scores are used (Eq. 3), while training the summarizer at a deeper layer, to decide the summary-worthiness of tweets. As the joint loss backpropagates through the shared encoder, BERTweet parameters are fine-tuned end-to-end to produce latent representations of tweets which benefit both the tasks.

\subsection{Scoring the Tweets}

For scoring, we consider the verifier-generated probability score \( V_{i}^{\text{prob}} \), and the summarizer-generated probability score \( \text{Summ}_i^{\text{prob}} \) (given by Eq. 5), for all those source tweets \( s_i \)’s where both the scores are greater than 0.5. We term these source tweets as relevant. Here we postulate that including tweets with high \( \text{Summ}_i^{\text{prob}} \) scores in our generated summaries would ideally help in achieving high ROUGE-F1 scores with respect to the gold-standard summaries, thereby ensuring high content-richness; whereas including tweets with high \( V_{i}^{\text{prob}} \) scores would help in improving the verified-ratio of summary tweets. In order to allow for this trade-off, we introduce a balancing parameter \( \kappa \) to define the relevance-score of the source tweet \( s_i \) as follows:

\[ \text{Score}(s_i) = \kappa \cdot \text{Summ}_i^{\text{prob}} + (1 - \kappa) \cdot V_{i}^{\text{prob}} \]

\subsection{Inference - Summary Generation}

For generating the final summary, our goal is to maximize the collective relevance-scores of the selected subset of tweets while minimizing redundancy. Let the desired length of summary be \( L \) words. We formalize these objectives as a constrained maximization problem using ILP. In the following equations, let \( l_i \) be the length (in words) of the source tweet \( s_i \). Let \( x_i \) be a binary indicator variable whose value indicates whether to include \( s_i \) in the final summary. \( \text{sim}_{ij} \) represents the cosine similarity between the embeddings (obtained using Sentence-Transformers [20]) of tweets \( s_i \) and \( s_j \). We use the Gurobi optimizer [4] to maximize the objective as defined in Eq. 9:

\[ \text{argmax} \sum_i \text{Score}(s_i) x_i - \sum_{ij} \text{sim}_{ij} y_{ij} \]

s.t. \( \sum_i l_i x_i \leq L \) (10)

\[ y_{ij} \leq \frac{1}{2} (x_i + x_j) \quad \forall \ i, j \]

\[ y_{ij} \geq x_i + x_j - 1 \quad \forall \ i, j \] (12)

Eq. 10 restricts the length of the summary to \( L \) words. Eqs. 11 and 12 ensure that \( y_{ij} \), another binary variable, is 1 if and only if both \( x_i \) and \( x_j \) are 1. Therefore, apart from maximizing the collective relevance of the selected subset of tweets, Eq. 9 tries to minimize their collective mutual similarity scores, thereby minimizing redundancy while enhancing the content-richness and the novelty of the information being covered. We therefore note that the final summary-membership of each relevant source tweet \( s_i \) is obtained considering the global context of all relevant candidate tweets in \( S \).

\section{3 EXPERIMENTS}

\subsection{3.1 Dataset and Annotation}

We use the PHEME dataset [29] consisting of 4,659 Twitter conversation threads, each with a source tweet and its replies, posted during four breaking-news events related to man-made disasters. For each thread, the source tweet is labeled as either rumour or non-rumour by a team of journalists. We consider the non-rumourous tweets as verified and rumourous tweets as unverified. Stance labels for a subset of rumourous threads were obtained from the RumourEval 2019 dataset [2]. Table 1 contains the dataset statistics.

For each of the four events, we obtain experts-curated extractive summaries (one per event) from Sharma et al. [25], all of which are around 250 words long. It is to be noted here that both the gold-standard as well as our model-generated summaries consist only of source tweets (and not replies) since the rumour/non-rumour annotations are only defined for the source tweets. Furthermore, all gold-standard summary tweets are non-rumourous/verified as per PHEME annotations. For each event, we thus have only around 20 tweets labeled as in-summary thereby making the dataset too skewed (towards the out-of-summary class) to efficiently train a supervised classifier for the summarization task. We therefore consider changing the labels of a few out-of-summary tweets (verified as per PHEME) to in-summary, based upon the cosine similarities of their embeddings with those of the original in-summary tweets. Tweet embeddings are obtained using Sentence-Transformers [20], a state-of-the-art approach for producing semantically meaningful sentence embeddings. The similarity threshold was empirically set
in the range of [0.7, 0.8] so that we could obtain a ratio of nearly 1.9 between in-summary and out-of-summary tweets for each of the events. While these updated labels are used only for training, ROUGE1-F1 scores of model-generated summaries are calculated with respect to the original gold-standard in-summary tweets.

### 3.2 Experimental Setup

We use the bertweet-base pre-trained model from the HuggingFace library to initialize the parameters of our shared encoder. In the verifier, hidden state dimensions of the Tree-LSTM cells are set to 128. As illustrated in [25], 80-90% of the replies are at levels ≤ 5. Hence, during training, the maximum height of propagation trees is set to 5. We note here however that our trained model can be safely applied to new events for determining the credibility (and summ ary-worthiness) of source tweets in near real-time without having to wait for their replies. Our codes are suitably adjusted to handle such cases. For training our summarizer, we use the default parameters as defined in [18]. Other model parameters are randomly initialized.

All our models, including the baselines, are trained using the Leave-one-out ($L$) principle, where in each run of the experiment, the data from one event is set-aside as test-set while the labeled tweets from all other events are used for training. This helps us to evaluate and compare the generalizability of our proposed solution. We set aside 10% of the training data as validation set for hyper-parameter tuning. While doing so, we maintain similar class distributions in the train and val sets for the corresponding tasks. For each train-test combination, we perform extensive experiments with different sets of hyper-parameters, select the model corresponding to the best validation performance, and evaluate it on the test set. Our final models are trained with learning rates ranging from $1e^{-6}$ to $1e^{-4}$, optimizer weight decays ranging from 0.1 to 0.001, and no. of epochs ranging from 5 to 20. All our models are trained end-to-end with Adam optimizer on Tesla P100-PCIE (16GB) GPU. Finally, we ensure the reproducibility of our results and make our codes and preprocessed datasets publicly available \(^1\).

### 3.3 Evaluation Metrics

From Table 1, we observe a considerable imbalance in verified and unverified classes in our dataset. Hence, for evaluating the rumour detection/verification performance, we use the macro-averaged F1 (or Macro-F1) scores. We however also report the Accuracy scores following prior works. For the evaluation of model-generated summaries, we consider the following metrics: (i) V-Ratio (verified ratio): representing the proportion of verified tweets in the summaries, and (ii) R-1: representing the ROUGE-1 F1 scores obtained against the gold-standard summaries (all 250 words long).

\(^1\)https://github.com/rajdeep345/MTLTS

### 3.4 Related Work and Baselines

#### Tweet Verification

Recent research on detecting rumours from social media conversations predominantly relies on tree-based neural classifiers [13, 15] to learn the temporal and structural properties of information cascades. [10, 14] additionally exploit the stances of community responses to learn rumour-specific signals. Different from these, multi-task learning has been used in [1, 9] to train several rumour-related subtasks. Recently, Cascade-LSTM (Ducci et al. [3]) and Min et al. [8] have addressed the limitations of tree-based approaches. Our verifier (Section 2.2.2) combines the advantages of Cascade-LSTM and stance-classification to eventually improve the rumour detection performance. We compare the performance of STLV (Fig. 1a) with the following representative state-of-the-art baselines:

- CETM-TL (Sharma et al. [25]): a bottom-up Tree LSTM-based classifier that leverages tweet features derived from an unsupervised LDA-based Content Expression Topic Model (CETM).
- TD-RvNN (Ma et al. [15]): a top-down Recursive Neural Networks-based rumour classifier.
- TL-Conv (Kumar et al. [10]): Tree LSTM with convolution units to jointly address rumour detection and stance classification.
- VRoC (Cheng et al. [1]): a variational autoencoder-based rumour classifier that jointly trains all four components of a rumour classification system as defined in [27]. For VRoC, we consider the results of its rumour detection component for our comparisons.
- Cascade-LSTM (Ducci et al. [3]): proposes a novel bi-directional variant of Tree-LSTM to accumulate the knowledge from the entire information cascade at each tree node.

#### Trustworthy Summarization

Studies on summarizing social media data are limited in comparison to the vast literature on document summarization [12]. Addressing this gap was one of our prime motivations. Accordingly, we leverage SummaRuNNer [18], and present a novel way to make it suitable for summarizing disaster-related tweets. Exploring the possibility of utilizing more recent extractive document summarization techniques for Twitter summarization is envisaged as a future work. Given the focus of our study, we compare the performance of MTLTS with the following disaster-specific summarizers:

- APSAL (Kedzie et al. [7]) is a clustering-based multi-document summarization system that focuses on human-generated information nuggets to assign salience scores to disaster-related articles.
- COWTS (Rudra et al. [22]) is an extractive summarization approach specifically designed to generate situational summaries from disaster-related tweets.
- SCC (Rudra et al. [24]) is an extractive summarization approach that first identifies sub-events from disaster-related tweets, followed by obtaining a summary for each identified sub-event. Taking cues from Sharma et al. [25], we identify four sub-events relevant to PHEME events (refer Section 5.3), obtain a summary for each sub-event, and report the average scores for comparison.
- VERISUMM (Sharma et al. [25]) is the existing state-of-the-art for this task. It takes a pipeline approach by first verifying the tweets using CETM-TL, followed by solving a constrained-maximization problem (using ILP) for jointly optimizing the credibility and content richness of the generated summaries.

### Table 1: Statistics of the PHEME Dataset.

| Disaster-Events          | # Tweets w/ Stance | # of Source Tweets | Verified | Unverified | Total |
|--------------------------|--------------------|--------------------|----------|------------|-------|
| Charlie Hebdo            | 1,071              | 1,621 (78.0%)      | 458 (22.0%) | 2,079      |
| Germanwings-crash        | 281                | 231 (49.3%)        | 238 (50.7%) | 469        |
| Ottawa shooting          | 777                | 420 (42.2%)        | 470 (52.8%) | 890        |
| Sydney siege             | 1,107              | 699 (57.2%)        | 522 (42.8%) | 1,221      |
| Total                    | 3,236              | 2,971 (64%)        | 1,688 (36%) | 4,659      |
We compare the results of MTLTS with our summaries generated by various methods. Overall, MTLTS achieves best Verified Ratio of summary tweets for $\kappa = 0$ and best ROUGE-1 F1 scores for $\kappa = 1$. MTLTS outperforms VERISUMM with 21 - 35% gains in V-Ratio. In terms of ROUGE-1-F1 scores, we outperform the strongest baseline SCC by a margin ranging between 2 - 5% gains.

| Model              | Charliehebo V-Ratio | Charliehebo R-1 | Germanwings V-Ratio | Germanwings R-1 | Ottawa V-Ratio | Ottawa R-1 | Sydney V-Ratio | Sydney R-1 | Overall V-Ratio | Overall R-1 |
|--------------------|----------------------|------------------|----------------------|------------------|----------------|-------------|----------------|-------------|----------------|-------------|
| APSAL [7]          | 0.200               | **0.493**        | 0.056               | 0.437            | 0.000          | 0.435       | 0.000          | 0.483       | 0.064          | 0.462       |
| COWTS [22]         | 0.385               | 0.479            | 0.111               | 0.504            | 0.118          | 0.427       | 0.087          | 0.479       | 0.175          | 0.472       |
| SCC [24]           | 0.365               | 0.465            | 0.244               | 0.505            | 0.212          | 0.444       | 0.308          | 0.492       | 0.282          | 0.477       |
| VERISUMM [25]      | 0.722               | 0.381            | 0.624               | 0.519            | 0.645          | 0.396       | 0.619          | 0.378       | 0.653          | 0.419       |
| w/ STLV            | 0.781               | 0.352            | 0.703               | 0.555            | 0.720          | 0.362       | 0.730          | 0.417       | 0.734          | 0.422       |
| Our Methods        |                      |                  |                      |                  |                |             |                |             |                |             |
| STLS               | 0.667               | 0.430            | 0.789               | 0.510            | 0.629          | 0.431       | 0.632          | 0.474       | 0.679          | 0.461       |
| + STLV             | 0.749               | 0.454            | 0.825               | 0.558            | 0.732          | 0.416       | 0.732          | 0.487       | 0.760          | 0.479       |
| MTLTS              |                      |                  |                      |                  |                |             |                |             |                |             |
| w/ $\kappa = 1$    | 0.750               | 0.444            | 0.927               | 0.565            | **0.860**      | **0.477**   | **0.627**      | **0.518**   | 0.791          | 0.501       |
| w/ $\kappa = 0.5$  | 0.689               | 0.420            | 1.000               | **0.582**        | 0.813          | 0.468       | 0.687          | 0.471       | 0.797          | 0.485       |
| w/ $\kappa = 0$    | **0.947**           | **0.472**        | **1.000**           | 0.575            | 0.834          | 0.464       | 0.750          | 0.452       | **0.883**      | 0.491       |

Table 3: Qualitative comparison of Rumour Detection/Tweet Verification results, averaged across all four train/test splits over the PHEME dataset.

| Model              | Accuracy | Precision | Recall | F1-Macro |
|--------------------|----------|-----------|--------|----------|
| CETM-TL [25]       | 0.727    | 0.715     | 0.695  | 0.704    |
| TD-RvNN [15]       | 0.737    | 0.748     | 0.738  | 0.743    |
| TL-Conv [10]       | 0.740    | 0.743     | 0.747  | 0.745    |
| VRoC [1]           | 0.752    | 0.755     | 0.752  | 0.752    |
| Cascade-LSTM [3]   | 0.768    | 0.762     | 0.756  | 0.739    |
| STLV               | 0.783    | 0.767     | 0.765  | 0.766    |
| MTLTS              | 0.786    | 0.770     | 0.766  | 0.768    |

We further perform a few ablations of our proposed architecture:

- STLS, as depicted in Fig. 1b, represents the summarizer when trained as a single task (without tweet credibility verification).
- STLS + STLV: We train STLS and STLV separately for each train-test combination. During inference, we use the tweet-level classifier probabilities, respectively generated by the two modules, for generating the final summaries.

### 4 MAIN RESULTS

#### 4.1 Evaluating MTLTS

We compare the results of MTLTS with our summaries generated by various methods and baselines for the joint task of trustworthy summarization in Table 2. Here, we remind our readers that CETM-TL, thereby demonstrating the better learning capabilities of a top-down tree-based classifier. TL-Conv, despite being a bottom-up approach, marginally outperforms TD-RvNN, which demonstrates the advantage of jointly learning stance detection with rumour classification. VRoC performs better than the previous methods (including STLV) are trained using the L principle (Section 3.2) for each train-test combination of PHEME events. Scores averaged over four runs of the experiment (each with one of the four events as the test set) are reported in Table 3. Results for VRoC are however taken from Cheng et al. [1]. MTLTS represents the verification scores obtained during the training of MTLTS.

From Table 3, we note that TD-RvNN comfortably outperforms CETM-TL, thereby demonstrating the better learning capabilities of a top-down tree-based classifier. TL-Conv, despite being a bottom-up approach, marginally outperforms TD-RvNN, which demonstrates the advantage of jointly learning stance detection with rumour classification. VRoC performs better than the previous

#### 4.2 Evaluating Tweet Verification

All methods (including STLV) are trained using the L principle (Section 3.2) for each train-test combination of PHEME events. Scores averaged over four runs of the experiment (each with one of the four events as the test set) are reported in Table 3. Results for VRoC are however taken from Cheng et al. [1]. MTLTS represents the verification scores obtained during the training of MTLTS.

From Table 3, we note that TD-RvNN comfortably outperforms CETM-TL, thereby demonstrating the better learning capabilities of a top-down tree-based classifier. TL-Conv, despite being a bottom-up approach, marginally outperforms TD-RvNN, which demonstrates the advantage of jointly learning stance detection with rumour classification. VRoC performs better than the previous
Table 4: Human-Eval. Results with 70 respondents. Values indicate (%) times a method is preferred for a given question.

| Methods    | CharlieHebdo | Germanywings | Ottawa    | Sydney    |
|------------|---------------|--------------|-----------|-----------|
|            | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
| COWTS      | 0.07 | 0.14 | 0.20 | 0.23 | 0.19 | 0.14 | 0.13 | 0.10 |
| SCC        | 0.16 | 0.19 | 0.10 | 0.13 | 0.11 | 0.12 | 0.10 | 0.14 |
| VERISUMM   | 0.15 | 0.14 | 0.10 | 0.12 | 0.12 | 0.11 | 0.13 | 0.14 |
| MTLTS      | 0.54 | 0.63 | 0.39 | 0.56 | 0.47 | 0.58 | 0.56 | 0.70 |

Table 5: Categories of crisis-related tweets in the PHEME dataset along with their corresponding Keywords.

| Class                  | Keywords obtained using CETM | Keywords obtained using CatE |
|------------------------|------------------------------|-----------------------------|
| Casualties             | anti-killed, murder, outcry   | anti-killed, murder, outcry |
| Police Investigation    | shootings, burst, crimes, suspects, videos, etc. | police, shootings, crime |
| Other Event-specific Updates | police, investigation, ongoing crisis | police, investigation, crime |
| Siege                  | hostages, family, condolences, family, thinking, sympathy, justice, sydney, ottawa, situation | hostages, family, condolences, family, thinking, sympathy, justice |
| Personal Opinions      | people, thoughts, sadness, sadness, feelings, sadness | people, thoughts, sadness |
| Public Opinions        | police, shooting, shot, killed, gunman, suspect, video, confirm, police, armed, cops, photo, gas, forces, justice | police, shooting, shot, killed, gunman, suspect |

Three methods since its rumour detection component is benefited from the simultaneous training of three other related tasks. The results of Cascade-LSTM well-establish the importance of learning bi-directional embeddings at each tree node. STLV combines the best of both worlds by extending the architecture of Cascade-LSTM to jointly train stance classification at each of the response nodes in the message propagation tree. We achieve new state-of-the-art performance on the PHEME dataset for the task of rumour detection, with our results being statistically significant than Cascade-LSTM with 95% confidence interval. MTLV results are comparable with those of STLV. Here, we would like to highlight that tweet verification plays the role of an auxiliary task in our case for eventually improving the performance of trustworthy summarization.

5 ANALYSIS AND DISCUSSIONS

5.1 Why Multi-task Learning?
In order to understand the scope of taking a multi-task approach, we first train the verifier and summarizer modules separately and report our results for STLS and STLV in Tables 2 and 3 respectively. STLS results highlight a considerable scope of improvement in the V-Ratio scores. For each train-test combination, we then explicitly use the verifier predictions to calculate the relevance-scores of tweets (following Eq. 8, with \( \alpha = 0.5 \)), before reconstructing the summaries using ILP. Considerable (overall 11.9%) gains in V-Ratio scores (row STLS + STLV in Table 2) prompted us to explore the advantage of training them jointly using a hierarchical multi-task setup, with trustworthy summarization being the main task, and tweet verification playing the role of an auxiliary task.

5.2 Human Evaluation Results
We perform a qualitative human evaluation of MTLTS-generated summaries (\( \kappa = 0 \)) against our baselines (all around 250 words long). Specifically, we created a survey that was divided into four sections corresponding to the four events under consideration. In each section, we randomly placed the summaries generated by three methods since its rumour detection component is benefited from the simultaneous training of three other related tasks. The results of Cascade-LSTM well-establish the importance of learning bi-directional embeddings at each tree node. STLV combines the best of both worlds by extending the architecture of Cascade-LSTM to jointly train stance classification at each of the response nodes in the message propagation tree. We achieve new state-of-the-art performance on the PHEME dataset for the task of rumour detection, with our results being statistically significant than Cascade-LSTM with 95% confidence interval. MTLV results are comparable with those of STLV. Here, we would like to highlight that tweet verification plays the role of an auxiliary task in our case for eventually improving the performance of trustworthy summarization.

5.3 MTLTS Summaries and Information

### Classes Specific to Man-made Disasters

Different stakeholders might be interested in different aspects of the ongoing crisis, such as information about victims, updates on police investigations, regions affected by the event, and general public sentiments, among others. Hence, it becomes crucial to analyze the extent to which MTLTS summaries capture trustworthy content across different classes of information specific to man-made disasters. We therefore need to first identify such classes relevant for the PHEME events. Based upon our observations and taking cues from [25], we consider the following four classes: (a) Victims and Casualties, (b) Police Investigation, (c) Other Event-specific Updates, and (d) Personal Opinions. In the absence of gold-standard labels, we use WestClass [17], a keyword-driven text classifier to predict the information class for each tweet. Performance of such a classifier however depends upon the discriminative nature of the embedding space it works with and the quality of weak-supervision provided to it by means of seeds/keywords for the expected categories/classes. We explore two methods here.

We first examine the top words obtained using the Content Expression Topic Model (CETM) [25] for each of the information classes identified above. From Table 5, we notice the existence of overlapping keywords across classes, for e.g., words such as "condolences" and "family" appear under both Victims and Casualties, and Public Opinions. This known limitation of such unsupervised LDA-based topic models makes the downstream text classification task difficult. We therefore leverage CatE [16], a recently proposed category-name guided text embedding method, that takes a set of category/class names as user guidance (with optional seed words for each class) to learn category distinctive word embeddings which help in retrieving a set of representative and discriminative terms under each given category. We train CatE with the information class names identified above along with their corresponding keywords obtained using CETM, and report the top keywords retrieved by CatE for each class in Table 5. The keywords are noticeably distinctive and semantically consistent with their corresponding class names (for e.g., names of victims and suspects being mapped to respective classes). We therefore continue with CatE for further analysis.

The discriminative word embedding space learnt using CatE is then used to train the WestClass text classifier with the same information class names provided to it as weak supervision. Distribution of verified and unverified tweets (as per PHEME annotations) across
all four information classes (using WestClass predictions) are reported in Table 6. We find that around 30% of the tweets (1,391 out of total 4,659) contain public opinions or non-situational content, out of which around 87% are verified. This observation led us to investigate if the improved V-Ratio scores of MTLTS summaries were owing to the inclusion of majorly non-situational tweets. We therefore generate the summaries for all four events with three separate values of \( \kappa \) (0, 0.5, 1), and report the percentage distribution of summary tweets across the four information classes in Table 7. Averaged over all summaries, only around 15% of the summary tweets change their veracity status to non-situational content, thereby leading us to the conclusion that in addition to capturing trustworthy information, MTLTS-generated summaries contain substantial amount of actionable or situational content.

Lastly, we examine the verified ratio of MTLTS summaries, generated separately for each class of information. Since rumorous claims posted during emergencies are eventually either confirmed or denied by reliable sources, we additionally investigate the eventual verified ratio (EVR) of our class-wise summaries using the veracity labels of tweets obtained from a subsequent version [30] of PHEME. We discover from Table 8 that many tweets originally marked as unverified become eventually verified across all classes. The summaries generated from tweets related to Police Investigations are especially less reliable in the beginning, despite containing potentially true information. As official details are gradually released by authoritative agencies, originally unverified tweets change their veracity status leading to a substantial improvement in verified ratio (EVR scores) for the same set of summaries for the class.

### 5.4 Case Study - Paris Terror Attack (Oct'20)

In order to evaluate the practical utility of our proposed system, we present a case study on a recent terror attack in the northwest suburb of Paris where a French teacher, who had shown his students cartoons of the Prophet Muhammad, was decapitated outside his school on 16th October 2020. The suspect was shot dead by the police minutes later. We scraped all the tweet threads available within three days from the date of the event, and used MTLTS (trained on all four datasets, \( \kappa = 0 \)) to generate a summary of length 250 words. 80% of the summary tweets were found (manually) to contain verifiable content, with the top 3 tweets, all conveying diverse information, along with their relevance scores shown below.

Table 7: Percentage distribution of MTLTS summary tweets across Information Classes specific to PHEME events. Values in brackets correspond to the % of tweets for a given class in the summaries for (Charlie Hebdo, Germanwings crash, Ottawa shooting, Sydney Siege).

Table 8: V-Ratio and Eventual Verified-Ratio (EVR) of Information Class-specific MTLTS summaries (with \( \kappa = 0 \)).

### 6 Conclusion

We present MTLTS, the first end-to-end solution for generating trustworthy summaries from large volumes of crisis-related tweets. MTLTS consists of two modules, a tweet summarizer and a rumour detector. Different from existing disaster-specific summarizers, we model tweet summarization as a supervised classification task. Also, in an attempt to address the gap between the vast body of literature on document summarization and comparatively limited research on social media summarization, we propose a novel way to modify the architecture of an existing extractive document summarization technique for summarizing tweets. For the auxiliary task of rumour detection, we achieve new SOTA results on the PHEME dataset, by jointly learning stances of replies towards the source tweet, and accumulating rumour-specific signals from the entire information cascade. Finally, we achieve new SOTA results for the primary task of trustworthy summarization by jointly training the two module in a hierarchical multi-task setup to exploit their interaction. While such an end-to-end thorough study lays the foundation, deploying our proposed solution to provide trustworthy updates in times of crisis would be one of our immediate future endeavours.

**ACKNOWLEDGMENTS**

This research is partially supported by IMPRINT-2, a national initiative of the Ministry of Human Resource Development (MHRD), India. Niloy Ganguly was partially funded by the Federal Ministry of Education and Research (BMBF), Germany (grant no. 01ID20003).

[T1]: https://en.wikipedia.org/wiki/Murder_of_Samuel_Paty
