Overview of the LT4CPR 2026 Shared Task: Generating Situation Reports from Social Media Messages

Fei Xia[1], **Haotian Zhu**[1], **Yun Woo Lee**[1],
Eitan Grenev[1], Allison K Burton[1], Yingzhe Fan[1],
William D. Lewis[1], Antonis Anastasopoulos[2], Belu Ticona[2]
[1] University of Washington, [2] George Mason University
[1]{fxia, haz60, yunu919, eitang, burton25, yingzf, wlewis2}@uw.edu
[2]{antonis, mticonao}@gmu.edu

Abstract

Language Technologies for Crisis Preparedness and Response (LT4CPR) encompass a broad range of natural language applications designed to support disaster response and preparedness. Within this domain, our shared task focuses specifically on leveraging social media data to extract actionable information and generate structured situation reports. To establish a formal benchmark, we introduce a domain-specific taxonomy for categorizing critical crisis information, alongside a novel pipeline for generating synthetic datasets to support task training and evaluation. Furthermore, we develop two baseline systems and propose an evaluation methodology designed to leverage the structural framework of situation reports. The shared task attracted participation from nine teams, whose diverse approaches offer valuable preliminary insights into the feasibility of structured situation report generation and the effectiveness of our evaluation setup. Finally, we highlight key open challenges (such as data access, operational alignment, noise filtering) which needs to be addressed in future research.

1 Introduction

There has been a significant amount of research on the use of Language Technologies (LT) to process crisis-related social media content. The first reported instance of social media being used during a crisis (Twitter, specifically) was during the 2007 San Diego firestorm (Meier, 2015). It was not until 2012, however, that tweets were processed automatically using LT and crowds during Typhoon Pablo, using NLP tools developed by the Stand-by Task Force (Liu, 2014). The Stand-by Task Force was formed in the wake of the 2010 earthquake in Haiti (Liu, 2014), and although social media was not used during that crisis, significant numbers of SMS messages were sent (up to 5,000 per hour) and processed using NLP tools developed by the SBTF and local partners (Mission 4636) which, when paired with a large Haitian Kreyòl speaking crowd provided with mapping tools, were responsible for saving hundreds of lives (Munro, 2013, 2010; Lewis, 2010).

It was the Haitian effort that made it clear that language technologies were life-changing and life-saving technologies, paving the way for a significant uptick in LT being applied to social media content. The first survey focused on the processing of social media messages during crises was Imran et al. (2015). Although Imran et al. (2015) looked at all aspects of using and processing social media content during crises (*e.g.,* information needs, social media monitoring, characteristics of crisis-related social media text, etc.), they did look at NLP as a set of tools necessary to pre-process content before acting upon it (*e.g.,* triaging, geo-tagging, geolocating). Subsequent work by many of the same authors saw the development of large Twitter-focused datasets through the Qatar Computing Research Institute (QCRI), such as Human Aid (Alam et al., 2021), Crisis MMD (Alam et al., 2018), TBCOV (Imran et al., 2021). etc. Having readily available, large datasets such as these led to a significant uptick in research focused on the application of LT to crisis-related social media text.

A recent survey (Lewis et al., 2025) examined 355 papers published in a five-year period from January 2020 to December 2024, all of which focused on processing crisis-related social media content. The survey finds that most of the LT systems (202 out of 292, 69.2%) built in this period concentrated on classification tasks, with much fewer systems on tasks such as NER (40), topic modeling (19), geo-location (11) and summarization (11).

The goal of our shared task is to promote and facilitate LT4CPR research that goes beyond these relatively simple tasks. For 2026, the shared task is to generate situation reports from social media messages. A situation report can be seen as a structured summary with strict requirements on what

Proceedings of the First Workshop on Language Technologies for Crisis Preparedness and Response, pages 203–213
November 9, 2026 ©2026 Association for Computational Linguistics

information should be included. Nine teams from six countries participated in the task.

This study makes several key contributions to the emerging LT4CPR field. (i) We establish a new shared task (Section 2), grounded in a novel taxonomy tailored specifically for LT4CPR (Section 3). (ii) We engineer a novel method for generating a synthetic dataset customized for the shared task (Section 4). (iii) Beyond conventional evaluation measures, we introduce and implement a fine-grained report-item-level comparison measure (Section 5). (iv) We construct two baseline systems and comprehensively evaluate their performance against participating systems (Section 6).

To foster open science and facilitate downstream research, we have publicly released all associated datasets, the custom LT4CPR taxonomy, software utilities, and evaluation scripts via the shared-task website.[1]

2 The Shared Task

The shared task focuses on transforming crisis-related social media streams into structured situation reports. During an ongoing crisis, social media posts can provide timely information on hazards, casualties, infrastructure damage, population movement, requests for assistance, response activities, and other developments. However, these streams are also highly redundant, fragmented, and sometimes uncertain or contradictory. Individual posts often provide only partial observations, while multiple posts may describe the same underlying development using different wording. The goal of the task is therefore to aggregate information distributed across multiple social media posts into a concise and structured representation of the crisis situation.

2.1 The Input: Tweets with Metadata

In this shared task, the input is a set of social media posts plus some metadata. For convenience, we refer to those posts as *tweets* but the posts can come from any social media platforms, not limited to Twitter. Each input is stored as a JSONL file. A tweet is represented as a record in the JSONL file. An example is as follows:

```
{
  "record_type": "tweet",
  "id": 42,
  "text": "Two reported injuries in Terminal 1",
  "information_source": "Media",
  "timestamp": "Tue Jul 08 19:57:44 +0000 2025"
}
```

[1] https://lt4cpr.github.io/aacl2026-workshop-LT4CPR/shared-task.html

The record has five fields:

- The first field, "record_type", always has the value "tweet", to distinguish the tweet lines from lines for metadata.
- The second field, "id", is an identifier used to link tweets in the input with the report items in the system output.
- The third field is the main content of the tweet. While it is "text" for this year's shared task, it can be changed to other modalities such as audio and images in future shared tasks.
- The fourth field specifies the type of information source, which can be used as a factor when determining how reliable the tweet is.
- The last field is the timestamp of the tweet.

The metadata includes fields such as *title* (e.g., "Eruption of Mount Vurren in the Sarnat Highlands"), *hazard type* (e.g., "volcano"), and a reporting period (e.g., "from 1am to 11pm PST on Sept 10, 2025"). Only tweets that were posted within the reporting period and were related to this particular crisis should be used to generate the system output.

2.2 The Output: A Situation Report

In this shared task, the output is called a situation report (**SitRep**), which must follow a predefined format.

Sections and Subsections: Conceptually, a SitRep is a document consisting of sections and subsections. The titles of sections and subsections are fixed, as shown in the second and third columns of Table 1. The first section is always called *Situation overview* and it is a short summary of the crisis and specifies (1) what happened, where, and when; (2) human impact such as casualty and displacement figures; (3) the principal cause or mechanism (if established), and (4) the response and the current status. The second section of a SitRep is always called *Timeline*. It lists important events in chronological order.

Sections 3-11 in the SitRep cover different aspects of the crisis, such as casualties, hazard assessment, and aid provided. Each section is further divided into subsections, and each subsection includes only a subsection title and a bulleted list.

Among the 11 sections, the system output must include Sections 1 and 2. Other sections and subsections are optional; they should be included in

ID	Section title	Subsection title	Event Superclass	Event Type
1	Situation overview	NA	NA	NA
2	Timeline	NA	NA	NA
3	Casualties and human impact	Deaths, Injuries, Missing persons, Rescues	person_state_change	KILL, INJURY, MISSING, RESCUE
4	Infrastructure and service impact	Destruction, Service disruption	object_infra_change	DESTROY, DISRUPT_SERVICE
5	Displacement and movement	Evacuations, displacement, returns	movement_displacement	EVACUATE, DISPLACE, RETURN
6	Hazard assessment	Severity, Development of the hazard	hazard_state	SITUATION_STATE, HAZARDOUS_CONDITION
7	Response actions	Operational response	response	DEPLOY, COORDINATE
8	Communication and information	Warnings, Information updates	communication	WARN, REQUEST, INFORM
9	Aid and relief	Aid provided, Aid received, Donation	provide_help, receive_help	PROVIDE_HELP, RECEIVE_HELP
10	Organizational and administrative activity	Meetings, economic measures	org_admin	MEETING, POLICY_CHANGE, FUNDING
11	Social and community response	Social support, Public sentiment	social, cognitive	SOCIAL_SUPPORT, ATTITUDE

Table 1: Sections in a situation report and corresponding event types. The first three columns show Section ID, Section title, and Subsection titles in a situation report. The fourth column shows the Event Superclass that normally corresponds to that Section Title, and the last column shows the Event Types that normally correspond to the Subsection Titles in the third column.

the system output if and only if the input tweets provide relevant information for those (sub)sections.

A Report Item: To avoid confusion, we call a bullet in a bulleted list in a SitRep a *report item*. A SitRep should be stored as a JSON file. Below shows how a report item in the *Injuries* subsection is represented in this format (Comments starting with "//" are not part of the format).

```
{
  ... // metadata about this SitRep
  "sections": [
  ...  // Section 1 and 2 are here
  {
    "id": "3", // section id
   "title": "Casualties and human impact",
    "subsections": [
    {
      "id": "3b", // subsection id
      "title": "Injuries",
      "bullets": [
      {
        "id": "3b.1",
        "text": "Four injured at LAX",
        "confidence": "confirmed",
        "tweet_ids": [12, 18, 42]
      } // end of the current bullet
      ... // other bullets
    ]  // end of bulleted list in Sect 3b
    } // end of Sect 3b
    ... // other subsections
  ] // end of subsections in Section 3
  } // end of section 3
    ... // other sections
 ] // end of sections
} // end of the sitation report
```

Connection between input and output: The *tweet_ids* field in a report item links input tweets with the report items in the SitRep. Thus, this shared task combines the traditional summarization task with evidence grounding and structured generation.

3 LT4CPR Taxonomy

In a situation report, each subsection serves to summarize multiple events of the same type. For instance, the *Injuries* subsection summarizes all the *INJURY* events mentioned in the input tweets.

Event identification (often formulated as event detection or trigger extraction) has long been established as a core task within Information Extraction and NLP (e.g., (Walker et al., 2006; Chen et al., 2015; Xiang and Wang, 2019; Lai, 2022)). Building upon this long line of research and designed to facilitate structured analysis of crisis-related messages, we propose a LT4CPR-specific taxonomy

for *entities* and *events*. Events additionally incorporate typed arguments, epistemic status, negation, and optional event-specific attributes. This representation provides a standardized, structured interface bridging raw tweet text with downstream cross-message aggregation and summarization.

3.1 Entities

The entity inventory includes categories such as people, organizations, locations, hazards, infrastructure, resources, and temporal expressions. Table 2 comprises seven top-level entity types. Type-specific attributes provide additional structure wherever appropriate. For instance, a person entity can have an attribute called "address", which itself is a "loc" entity.

3.2 Events

The taxonomy organizes crisis event types hierarchically. Its superclasses encompass the major categories of crisis information, including changes in person state, infrastructure damage, movement and displacement, etc. More granular event types are defined beneath these broader classes (refer to the fourth column in Table 1).

In the event detection task, an event has a specific type, a textual trigger, zero or more event arguments, and a series of event attributes. Event arguments represent the participants and contextual information associated with an event. Our taxonomy uses PropBank-style argument labels together with event-specific semantic definitions. Core roles include `ARG0`, `ARG1`, and `ARG2`, while adjunct roles encode information such as location, time, cause, source, destination, manner, and purpose. For event attributes, the most important ones for us are *Realis* and *Negation*. Realis captures how the event is presented with respect to occurrence, uncertainty, planning, prediction, or conditionality. Negation indicates whether an event is negated.

3.3 Usage of the taxonomy

Our shared task leverages this taxonomy across multiple dimensions. First, we use it to define what each section and subsection in the situation report should cover. In general terms, an event superclass maps to a section and an event type maps to a subsection in the situation report, but the mapping is not strictly bijective (see Table 1); for instance, Section 1 or 2 does not correspond to a single event superclass. Second, we use the taxonomy in our synthetic dataset generation pipeline (Section 4) and our baseline systems (Section 6). Third, we release this taxonomy to shared task participants, who can choose to use or not use it in their participating systems.

4 The Dataset

Processing social media in mass emergencies is a well-established research domain (Imran et al., 2015), and underpinned by a sizable number of datasets and evaluations. For instance, a recent survey (Lewis et al., 2025) reveals that over 350 papers on processing crisis-related social media data were published between January 2020 and December 2024, but the vast majority of corpora built during that period were text-only, English-only, and sourced exclusively from Twitter. Because many disasters occur in non-English speaking countries, the value of creating resources for non-English languages cannot be overstated, especially if these resources are intended for real-world use.

However, owing to structural shifts in the social media landscape over the past several years, accessing or releasing social media text has become increasingly prohibitive. Notably, access to Twitter/X content has been severely restricted—either through restrictive paywalls or the mandate to "rehydrate" content (meaning only pointers to tweets can be shared, rather than the raw text itself)—thereby drastically reducing the volume of readily available content. Without accessible text corpora, data collection and tool development have grown significantly more difficult. Even when open collections of social media data can be located, generating gold-standard situation reports for them remains exceptionally challenging.

To address these persistent obstacles, we developed a robust framework that produces synthetic data tailored for the shared task.

4.1 Dataset Generation

The generation pipeline is anchored in a foundational principle: *the semantic plan precedes the post*. Instead of generating tweets first and situation reports subsequently, we begin with a concise narrative description of the crisis. The pipeline proceeds through several steps:

1. Select a hazard type (e.g., "floods") and instantiate an instance of that type by randomly populating the hazard attributes (e.g., "flooding at Region X that lasts 10 days").

Entity Type	Attribute	Attribute values
person	gender	male, female, etc.
	age_group	child, youth, adult, elderly, unknown.
organization	org_type	government, ngo, emergency_service, company, military, unspecified_organization
hazard	hazard_type	Hierarchical natural, human-induced, health, conflict, utility, and unspecified hazards.
infrastructure	infra_type	Transportation, utility, and public-service infrastructure.
resource	resource_type	food, water, electricity, shelter, fuel, medical, phone, internet, other
time	time_type	absolute, duration, recurring, unknown
loc	loc_type	geopolitical locations, natural locations, and addresses/intersections

Table 2: Top-level entity types and their attributes in the LT4CPR Taxonomy

2. Randomly generate report items that align with the hazard type (e.g., "15 people were injured", "100 people are missing"; "Several bridges collapsed").

3. For each report item, randomly determine the number of events that are relevant to that item. For example, for the item "15 people were injured", establish whether the injuries occurred within a single INJURY event or across multiple distinct ones (e.g., 3 injured at school Y, 4 at home, 8 on the highway).

4. For each event, we synthesize multiple tweet segments that explicitly reference the event. These segments may vary in their lexical realization of triggers, the inclusion or omission of event arguments and attributes, overall syntactic phrasing, etc. (e.g., "Oh, my god. Three people died today" vs. "several possible fatalities in Chicago").

5. Construct complete tweets by combining one or more tweet segments, ensuring that the resulting messages exhibit coherent linguistic styles. Furthermore, we enrich each tweet by incorporating metadata such as relevant hashtags, assigning a precise timestamp, and allocating a unique identifier.

6. Compile a gold situation report by appending tweet IDs to corresponding report items, organizing report items into structured bulleted lists, and adding section and subsection titles.

Because several of these phases entail stochastic processes, the pipeline can be executed iteratively to scale the dataset. In practice, since certain pipeline stages necessitate LLM API calls (e.g., Step 5, to enhance the natural fluency of the generated tweets), we reduce API costs by permitting different instances in the dataset to share selected tweets or report items.

4.2 Hazard Subtypes

The hazard entity type is defined hierarchically in our LT4CPR taxonomy and the top level comprises:

- **Natural hazards**: geophysical, hydrological, meteorological, and climatological hazards;
- **Human-induced hazards**: chemical spills, toxic releases, gas leaks, oil spills, nuclear incidents, radiation leaks, explosions, industrial fires, and hazardous waste;
- **Health hazards**: epidemics, pandemics, disease outbreaks, and environmental exposure;
- **Conflict hazards**: armed conflict, civil unrest, and terrorism; and
- **Utility hazards**: power outages, grid failures, dam failures, and bridge/building collapse risk.

Natural-hazard leaf nodes encompass, among others, earthquakes and aftershocks, tsunamis, volcanoes, landslides, avalanches, floods, thunderstorms, blizzards, hurricanes, typhoons, cyclones, tornadoes, heavy rain, strong winds, wildfires, droughts, heatwaves, and cold waves.

4.3 Corpus statistics

Table 3 summarizes the statistics of the generated dataset. The hazard subtypes for training, dev, and test sets are mutually exclusive (non-overlapping).

While the dataset is modest in scale, it can readily be expanded across multiple dimensions—including size, linguistic diversity (multilinguality), modality, and other factors such as data noise and the presence of conflicting information.

	train	dev	test
# of hazard subtypes	8	2	6
# of instances	140	23	117
Avg # of tweets/instance	283	372	104

Table 3: Statistics of the shared-task dataset. Hazard subtypes are discussed in Section 4.2.

5 Evaluation Measures

To evaluate a system, we compare its output against the gold standard. Ideally, the two should exhibit identical structural organization with a corresponding sequence of section and subsection titles. However, recall that Section 3-11 in a SitRep and their respective subsections should be present only if the input tweets have relevant events; thus, the system output and the gold standard could end up having different structures if the gold standard contains errors or if the system errs in identifying events or fails to strictly adhere to the output format requirements. Therefore, several methodological decisions are required for evaluation.

Skip two sections: We elect to ignore the first two sections in the SitRep because establishing a reliable gold standard for those sections requires more studies (e.g., what events should be treated as important and thus be included in the Timeline section).

Measures for comparing plain text: To compare two pieces of plain text, we utilize several well-known measures, including ROUGE-1, ROUGE-2, and ROUGE-L (Lin, 2004), BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020).[2]

Unit of comparison: Text can be compared hierarchically at four different levels: document level (where all structural properties in the reports are ignored), section level (where subsection boundaries are ignored), subsection level (where bullet boundaries are ignored), and bullet level (aka item level, which provides finer granularity).

For the shared task, we apply ROUGE, BERTScore, and BLEURT at the subsection level and then micro-average the results to derive a score at the test instance level.

We also implement an item-level evaluation leveraging the Hungarian Algorithm (Kuhn, 1955) to find an optimal alignment between report items in the system output and the gold report (see Appendix A).

Primary Score: For system-level results, we compute the mean of the scores for test instances within the same hazard and subsequently compute the mean across all hazards to ensure equal weighting across all hazards.

The official primary score is defined as:

$$\text{Primary} = \frac{\text{BERTScore}_{F_1} + \text{BLEURT}}{2}. \quad (1)$$

This score emphasizes semantic similarity notwithstanding variations in wording. ROUGE and item-level alignment are reported as complementary measures and are excluded from the official primary score calculation.

6 Baselines and Participating Systems

In this section, we describe two baseline systems we created for the shared task. Both systems integrate heuristics and Large Language Models (LLMs).

6.1 Baseline System #1

The first baseline uses a structured three-stage pipeline to process a collection of crisis-related tweets and generate a situation report.

Stage 1 runs an LLM (GPT-5.2) on individual tweets to identify entities and events in the tweets. The taxonomy outlined in Section 3 is used to formulatethe prompt for the LLM.

Stage 2 performs cross-tweet coreference by clustering entities (and events) identified in Stage 1 so that the objects in each cluster ideally refer to the same object in the real world.

Stage 3 uses event clusters from Stage 2 to generate a situation report. A section or subsection with no qualifying events is omitted.

6.2 Baseline System #2

The second baseline generates reports by LLMs and its main focus is on examining how the degree of preprocessing in the input affects situation-report generation.

Figure 1 shows the pipeline of the system. There are five options for the input: Level A consists of input tweets only; Level B adds the LT4CPR taxonomy; Levels C through E add the output produced by Baseline #1's Stage 1 or 2 (see Fig 2).

A template-based prompt generator fills a template with the input. It places the relevant input

[2]We use `microsoft/deberta-xlarge-mnli` for BERTScore and BLEURT-20 for BLEURT.

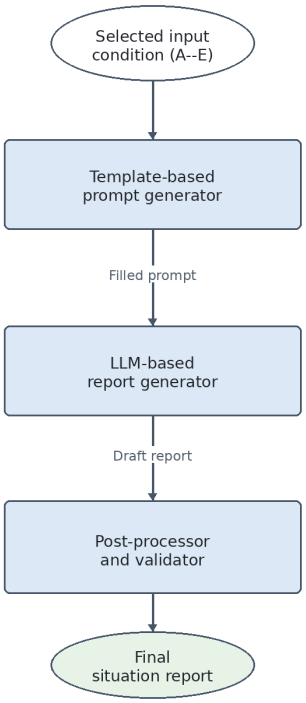


Figure 1: Common pipeline for all five conditions. Data products are shown as ellipses and processing modules as boxes; labels on arrows denote intermediate outputs.

alongside the reporting instructions and includes the taxonomy whenever the condition calls for it. The filled prompt is sent to an LLM to generate a draft report.

The draft report is then sent to a deterministic post-processor, which brings the response into the required report structure, resolves duplicated or misordered sections, and discards empty material. The processor then checks whether every cited tweet identifier occurs in the current test input. A rule-based validator finally checks the report schema, section names, confidence labels, and citation requirements.

6.3 Participating systems

Nine teams participated in the shared task; six of them submitted system description papers afterwards, which are included in the LT4CPR workshop proceedings. Most systems follow a similar pipeline that includes tweet classification, event extraction from individual tweets, clustering of tweets, entities or events, report item generation and verification, and report section serialization.

7 Results

Nine teams participated in the shared task. They were permitted to submit multiple systems but were required to designate one as their primary submission. One team submitted four systems, one submitted three, and two teams each submitted two systems. The remaining five teams each submitted only one system. In total, 16 systems were submitted.

7.1 Primary Submission Results

Table 4 reports the primary submission results for all participating teams, alongside the two UW baseline systems. The rows are sorted by the primary metric. Unsurprisingly, the ROUGE scores tend to be lower than those of BERT, BLEURT, and the Primary metrics, as ROUGE relies on exact n-gram matching.

The numbers in the *Align* column tend to be lower than in the other columns because it evaluates performance at the report-item level rather than the subsection level. Despite these differences, all metrics yield consistent system rankings.

7.2 Results of Best System per Team

Table 5 reports the highest-scoring submission from each participating team, regardless of whether it was designated as their primary submission. Notably, ranking teams by their best-performing submission yields an identical ordering to the primary submission leaderboard.

To evaluate whether performance gaps between adjacent rows in Table 5 are statistically significant on the test data with 117 instances, we conducted paired bootstrap and permutation tests on the primary scores. Pairs exhibiting a statistically significant difference at $p < 0.01$ are indicated with double asterisks on the higher-ranked system. Only four of the ten adjacent pairs show statistical significance, indicating that the score differences between closely ranked systems fall largely within the margin of statistical variation.

7.3 Performance of Baseline #2

The performance of Baseline #2 is comparable to the top participation systems. Table 6 illustrate its performance with five types of input and three LLMs: Gemini 3.8 Flash (**?**), GPT-5.6 Luna (**?**), and Claude Haiku 4.5 (**?**).

Among the three LLMs, Gemini produced the best results than the other two models except for

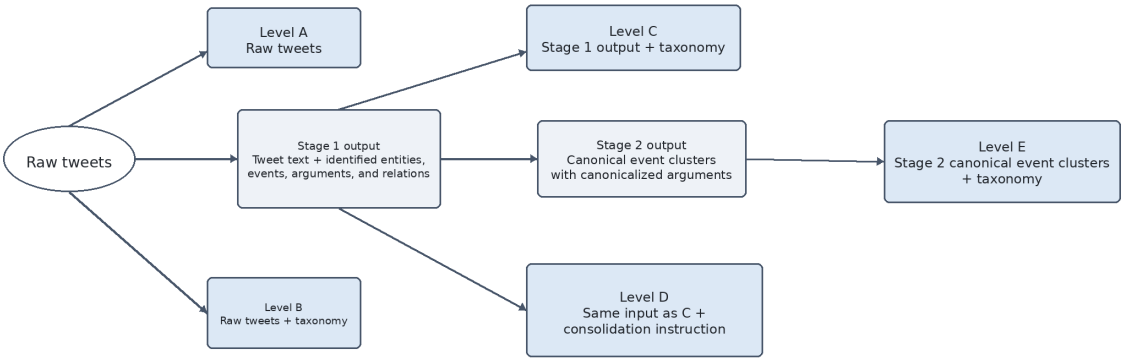


Figure 2: Derivation of the five input conditions. C and D receive the same Stage 1 data; D differs only by its consolidation instruction. E receives canonical event clusters produced by Stage 2.

Team	System	R1	R2	RL	BERT	BLEURT	Align	Primary
Stockmark	report-refinement	**0.6158**	**0.4481**	0.5486	**0.7269**	**0.6362**	**0.4635**	**0.6816**
UTS	v6	0.5820	0.4156	0.5331	0.7009	0.6275	0.4434	0.6642
UW Baseline #2	Level A (Gemini)	0.5988	0.4359	**0.5513**	0.7210	0.6000	0.4569	0.6605
Dy	rules-v2	0.5338	0.3684	0.4873	0.6739	0.6265	0.4056	0.6502
Meng-qiu in Zhongguancun	grounded	0.5300	0.4073	0.4871	0.5839	0.6071	0.3557	0.5955
Amrood	nemotron-120b-sitrep	0.5142	0.3927	0.4790	0.5984	0.5483	0.3715	0.5733
gogogo	r01g-sr	0.4645	0.3585	0.4398	0.5018	0.5528	0.2881	0.5273
UW System 1	Without grouping	0.3835	0.2129	0.3276	0.5297	0.5135	0.2331	0.5216
BMU	system_1	0.3594	0.2636	0.3219	0.4511	0.5170	0.2388	0.4841
Neural Coder	extraction-de	0.4331	0.3385	0.4127	0.5082	0.4514	0.3278	0.4798
Databaes	g5-evidence-grounded	0.2846	0.1715	0.2524	0.3947	0.5153	0.2022	0.4550

Table 4: Results of the teams' **primary** submissions and the two UW baselines, ordered by official Primary score. The highest score in each column is in boldface. R1, R2, RL, BERT, and Align denote ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, and item-level aligned F_1, respectively. Primary is the mean of BERTScore F_1 and BLEURT.

Level E. Among five levels of input, the level that produces the highest score varies from LLM to LLM.

8 Challenges

This shared task highlights many challenges that our LT4CPR research still faces.

Data access: The first is access to social media data, which has become increasingly constrained due to evolving platform API policies, heightened privacy regulations, and shifting data governance frameworks. These restrictions severely impede researchers' ability to collect localized, real-time text streams during unfolding emergencies, making it difficult to train and evaluate models on real-world crisis data.

Community need: Another challenge is to determine what the crisis response community actually needs on the ground. Without a clear understanding of the operational requirements of first responders and humanitarian organizations, computational models risk optimizing for generic language targets rather than actionable intelligence. Establishing these grounded practitioner needs is crucial, as it directly guides how system outputs should be formatted and dictates the specific metrics against which these technologies must be evaluated.

System design The third challenge relates to system architecture and model design. Social media data is inherently noisy, uncurated, and frequently inundated with conflicting statements or deliberate misinformation. To be effective, an LT4CPR system must incorporate robust filtering mechanisms capable of isolating critical situational signals from background noise and verifying factuality under time-sensitive conditions. Furthermore, systems must gracefully reconcile conflicting reports that arise during rapidly evolving emergency situations.

A related core difficulty lies in event coreference and entity resolution—specifically, accurately identifying and aggregating disparate posts that refer to the exact same real-world incident despite varia-

Team	System	R1	R2	RL	BERT	BLEURT	Align	Primary
Stockmark	report-refinement	**0.6158**	**0.4481**	0.5486	**0.7269**	**0.6362**	**0.4635**	**0.6816****
UTS	v6mbr	0.5770	0.4089	0.5300	0.7114	0.6221	0.4440	0.6667
UW Baseline #2	Level A (Gemini)	0.5988	0.4359	**0.5513**	0.7210	0.6000	0.4569	0.6605
Dy	rules-v2	0.5338	0.3684	0.4873	0.6739	0.6265	0.4056	0.6502**
Meng-qiu in Zhongguancun	cue	0.5300	0.4073	0.4871	0.5841	0.6071	0.3563	0.5956
Amrood	nemotron-120b-sitrep	0.5142	0.3927	0.4790	0.5984	0.5483	0.3715	0.5733**
gogogo	r01g-sr	0.4645	0.3585	0.4398	0.5018	0.5528	0.2881	0.5273
UW Baseline #1	Without grouping	0.3835	0.2129	0.3276	0.5297	0.5135	0.2331	0.5216**
BMU	system_1	0.3594	0.2636	0.3219	0.4511	0.5170	0.2388	0.4841
Neural Coder	extraction-de	0.4331	0.3385	0.4127	0.5082	0.4514	0.3278	0.4798
Databaes	g5-evidence-grounded	0.2846	0.1715	0.2524	0.3947	0.5153	0.2022	0.4550

Table 5: Results of each participating team's **best** system and the UW baselines. The highest score in each column is shown in bold. Significance markers compare each system with the immediately following system, using the larger of the paired bootstrap and permutation p-values. ** denotes $p < 0.01$. Displayed scores use the official hierarchical aggregation, whereas significance tests use per-instance Primary scores.

System	A	B	C	D	E
Gemini	**66.1**	65.2	65.2	66.0	59.6
GPT-Luna	62.9	61.3	63.0	**64.2**	59.3
Claude	61.5	62.7	60.9	62.4	**63.4**

Table 6: Performance of Baseline #2 with three LLMs and five types of input (in primary score, multiplied by 100). A-E refers to Level A-E, as defined in Fig 2. The highest score in each row is in boldface.

tions in user phrasing, terminology, and contextual noise.

Evaluation measures: There is the need for improved evaluation metrics. While current methods leverage the structural format of situation reports, plain-text comparison still largely relies on n-gram metrics like ROUGE or embedding-based scores such as BERTScore. However, these metrics often fail to capture critical factual precision. A question-answering (QA)-based approach focused on specific report items—such as verifying whether "five people were killed"—offers a more plausible and reliable way to measure factual accuracy.

9 Conclusion

In this overview paper, we presented the inaugural LT4CPR shared task on automatic situation report generation from crisis-related social media streams. By introducing a domain-specific event taxonomy, a systematic synthetic data generation pipeline, multiple baseline architectures, and a structured evaluation methodology, we established a rigorous benchmark for evaluating natural language generation systems in high-stakes emergency contexts.

The participation of nine teams and the submission of sixteen diverse systems demonstrate both the feasibility of structured situation report generation and the vibrancy of the research community surrounding crisis informatics. Experiments show all measures produce almost identical system rankings and the score gaps between closely ranked systems are largely statistically non-significant.

Looking forward, key challenges persist across the field. Future work must address constrained access to real-time social media data, align model outputs more closely with the operational needs of first responders, build systems capable of filtering noise and resolving conflicting misinformation, and adopt fact-oriented evaluation frameworks—such as QA-based metrics—that go beyond traditional n-gram and embedding similarity. Addressing these open problems will be vital for deploying trustworthy, real-world language technologies in emergency management.

Ultimately, we hope this shared task and its accompanying resources will catalyze continued innovation at the intersection of language technologies and crisis preparedness and response, fostering tools that are not only technologically advanced but also operationally dependable for emergency responders and affected communities.

Acknowledgments

This work was supported in part by National Science Foundation under Grant No. CNS-2346335. We want to thank all the participanting teams for their support. We would also like to extend our gratitude to Yuqing Guo, Nicole N Mihaylova, and Margot Kleinpeter for their valuable contributions.

References

Firoj Alam, Ferda Ofli, and Muhammad Imran. 2018. CrisisMMD: Multimodal Twitter Datasets from Natural Disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*.

Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021. Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks. In *Proceedings of the 15th International AAAI Conference on Web Social Media (ICWSM)*, pages 933–942. AAAI.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176. Association for Computational Linguistics.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing social media messages in mass emergency: A survey. *ACM Comput. Surv.*, 47(4).

Muhammad Imran, Umair Qazi, and Ferda Ofli. 2021. Tbcov: Two billion multilingual covid-19 tweets with sentiment, entity, geo, and gender labels. arXiv:2110.03664.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97.

Viet Dac Lai. 2022. Event extraction: A survey.

William Lewis, Haotian Zhu, Keaton Strawn, and Fei Xia. 2025. Tapping into Social Media in Crisis: A Survey. In *Proceedings of the 4th Workshop on NLP for Positive Impact, EMNLP 2025*, vienna, Austria.

William D. Lewis. 2010. Haitian Creole: How to Build and Ship an MT Engine from Scratch in 4 Days, 17 Hours, & 30 Minutes. In *Proceedings of the 14th EAMT*, Saint Raphaël, France.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Sophia Liu. 2014. Crisis crowdsourcing framework: Designing strategic configurations of crowdsourcing for the emergency management domain. *Computer Supported Cooperative Work (CSCW)*, 23:389–443.

Patrick Meier. 2015. *Digital Humanitarians*. CRC Press, Boca Raton.

Robert Munro. 2010. Crowdsourced Translation for Emergency Response in Haiti: The Global Collaboration of Local Knowledge. In *AMTA Workshop on Collaborative Crowdsourcing for Translation*.

Robert Munro. 2013. Crowdsourcing and the crisis-affected community: lessons learned and looking forward from mission 4636. *Journal of Information Retrieval*, 16.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus.

Yue Xiang and Houfeng Wang. 2019. A survey of event extraction technology. *IEEE Access*, 7:173111–173137.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Item-Level Evaluation

For item-level comparison, we first break each situation report into a set of subsections and align them using subsection titles.

For a subsection d, let G_d and S_d denote the bulleted list from the gold report and the system output, resp. To calculate the similarity between G_d and S_d, we need to start with the similarity w_{ij} between the i gold item and the j system item. Let R_{ij} be their ROUGE-L F-score and J_{ij} the Jaccard similarity of the two items' tweet-id sets. Now we can define w_{ij} as a linear combination of the two scores:

$$w_{ij} = \lambda * R_{ij} + (1 - \lambda) J_{ij}. \quad (2)$$

For the shared task evaluation, we set λ to be 0.8 to give more weights to the ROUGE score.

Now we have calculated s_{ij} for all the (i, j) pairs, we run the Hungarian Algorithm (Kuhn, 1955) to find an optimal alignment that maximizes the total weights of the aligned items.[3]

Let W_d be the total weight of the optional alignment for subsection d. Let m and n be the number of report items across all the subsections in the gold

[3]From the optimal alignment, we retain only pairs with $w_{ij} > 0.1$ to discard negligible assignments rather than enforce strict semantic equivalence. This preserves partial credit for related items while treating weak matches as unmatched.

report and the system report, resp. Precision and recall are defined as follows:[4]

$$P = \frac{\sum_d W_d}{n}, \quad R = \frac{\sum_d W_d}{m} \tag{3}$$

The resulting F1 is the item-level score for the system report.

[4]According to this definition, if a gold item i and a system item j are connected according to the optimal assignment, the edge is considered a partial match and its score w_{ij} is added to the total match score.