Extractive Corroboration Pipeline for Crisis Situation Report Generation from Social Media

Sreeja Guha Majumdar
Heritage Institute of Technology
sreeja.guhamajumdar.csbs28@heritageit.edu.in

Abstract

Automatic generation of crisis situation reports from social media requires transforming noisy, short-form text into structured, evidence-grounded prose, without access to information outside a single bounded time window. This paper describes an extractive pipeline submitted to a shared task on this problem: a supervised section classifier trained on tweet-to-bullet citation links, corroboration-based confidence estimation, safety-netted extractive bullet-text synthesis, and rule-based temporal filtering. The submitted system passes the task's structural validator on all official test cells, spanning crisis types absent from training, and receives an official evaluation score of 0.480. Decomposing the two metrics composing this score shows that the system obtains a BERTScore F1 of 0.508 and a BLEURT score of 0.451, exposing a substantial difference between semantic similarity and learned fluency/adequacy evaluation. Development-time diagnostics, including a coreference-recall experiment and a per-section quality breakdown, provide independent evidence that extractive text generation is a major remaining limitation. Three architectural alternatives tested during development are reported with their measurements rather than omitted, and a scoped, safety-netted rewrite step is identified as a direct next architectural extension.

1 Introduction

Crisis situation reports translate an unfolding disaster into a structured document organized by topic — casualties, infrastructure impact, displacement, hazard assessment, response actions, and related categories — so that a decision-maker does not have to read the underlying social media stream directly. The shared task addressed here frames automatic report generation as a per-cell, per-window problem: given only the tweets posted during one bounded time window of one crisis event, a system must independently produce a report whose bullets are (a) grounded in cited tweet evidence, (b) labeled with a confidence level derived from corroboration, and (c) free of information the window could not plausibly have supported, such as retrospective framing or figures dated beyond the window's end.

Two properties distinguish this task from related extractive crisis-summarization benchmarks. First, gold bullet text is authored independently of any single tweet, drawn from a separately written full report rather than pasted from source material; a system that reproduces a representative tweet verbatim, however well selected, is measured against a sentence it did not generate. Second, systems process every cell independently, without access to other windows or other cells of the same event, bounding how closely a corroboration-based confidence estimate can approximate the task's full-report corroboration rule.

An earlier keyword-based baseline scored precision 0.044, recall 0.040, and F1 0.040 on the internal development metric (Section 5.1). Its limitations were attributed largely to three compounding causes: the extractive-text gap above; a hand-written keyword vocabulary calibrated to wildfire, flood, and earthquake crises with no coverage for crisis types such as chemical spills or structural collapses; and a labeled dataset — direct tweet-to-bullet citation links present in every gold report — left unused. This paper documents a replacement pipeline, submitted as system `extraction-de`, addressing each cause directly, evaluated against both the internal metric and the shared task's official evaluation.

The contributions of this paper are: (1) a supervised section classifier trained on citation-derived weak supervision, evaluated under a cross-crisis-type generalization protocol matching the task's own train/development/test structure; (2) a bullet-generation procedure combining corroboration-based confidence, latest-figure resolution, rule-

Proceedings of the First Workshop on Language Technologies for Crisis Preparedness and Response, pages 156–165
November 9, 2026 ©2026 Association for Computational Linguistics

based temporal filtering, and two safety nets — numeric grounding and negation preservation — guarding a future abstractive rewrite step against hallucination and polarity flips; (3) three architectural alternatives tested during development, reported with measurements rather than omitted; and (4) an analysis combining official results with development-time diagnostics to identify clustering granularity and text generation, rather than classification, as the main remaining sources of error.

2 Related Work

CrisisFACTS (McCreadie and Buntain, 2023) recasts the TREC Incident Streams track as a multi-stream (Twitter, Reddit, news, Facebook) crisis fact-extraction and timeline-summarization task, scored against near-verbatim retrieved-and-ranked source snippets — an extractive evaluation. This is a materially different task shape from the one addressed here, where gold bullets are independently authored sentences rather than retrieved snippets, which is one reason an extractive pipeline faces a structural text-generation challenge here that is less central in a source-snippet retrieval setting.

Crisis2Sum (Seeberger and Riedhammer, 2024) compares extractive and LLM-based abstractive baselines for multi-stream disaster summarization on the CrisisFACTS data, finding that an extractive, query-focused framework can perform competitively with abstractive approaches on long, information-dense summaries. This supports classify→cluster→fuse as a credible general architecture, matching the pipeline in Section 4.

On classification, zero-shot prompting of instruction-tuned language models has been evaluated on crisis-tweet classification (McDaniel et al., 2024) and compared against parameter-efficient fine-tuning on the HumAID benchmark (Han et al., 2026), where LoRA fine-tuning reaches roughly 80% accuracy against roughly 42% for zero-shot prompting on a comparable task. This motivates the supervised classifier in Section 4.2 over a zero-shot approach; a zero-shot fallback is nonetheless implemented and left disabled (Section 4.7).

For report generation, an uncertainty-aware framework (Delos Reyes and Kitamoto, 2025) composes situational-awareness classification, uncertainty assessment, and generation, cross-checking model-elicited claim plausibility against an external proxy (USGS PAGER earthquake-impact estimates). The present pipeline shares this classify/estimate-confidence/generate shape, but its confidence stage is corroboration-counting rather than model-elicited, and no external proxy is available under the cell-independence constraint, which disallows looking beyond the cell being scored.

For automated evaluation, BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020) are widely used to assess semantic overlap and natural language generation quality. In social media text mining, specialized BERT architectures such as CrisisBERT (Liu et al., 2021) and benchmarks like HumAID (Alam et al., 2021) demonstrate the value of domain-adapted representation learning for extracting humanitarian information under distribution shifts across disaster types. Finally, work on factual consistency in abstractive generation (Kryscinski et al., 2020) emphasizes numeric and logical faithfulness, motivating our explicit safety-net checks.

3 Task and Data

The shared task provides train, development, and test splits, each composed of independent cells: one crisis, one time window, one replicate. Development covers two crisis types (chemical release, earthquake) absent from training; test covers six further types (structural collapse, dam safety, heatwave, industrial fire, landslide, tornado) absent from both. This nested unseen-type structure measures generalization at every stage, not only at final test time.

Each gold report has eleven sections, nine scored (casualties, infrastructure, displacement, hazard assessment, response actions, communication, aid, organizational activity, social/community response); an overview section and a timeline are not scored. Each gold bullet carries a confidence label (confirmed/unconfirmed) and supporting tweet identifiers. Four rules govern scoring: a predicted bullet needs ≥ 1 supporting tweet (R1); its confidence should reflect corroboration strength (R2); it must not state a figure beyond what the cell's tweets support, using the most recently reported value for a repeatedly-updated quantity (R3); and, unless the cell is the crisis's final window, it must not cite an explicit date after the window's end (R3c) or use retrospective framing (R3d).

4 System Description

Figure 1 summarizes the pipeline. Each cell is processed independently end to end, satisfying the

task's independence requirement by construction rather than by post-hoc filtering. The stages are: section classification (§4.2), within-section clustering and confidence estimation (§4.3), bullet-text generation with safety nets (§4.4), temporal filtering (§4.5), and report serialization (§4.6).

4.1 Weak Supervision from Citation Links

Every gold bullet carries a `tweet_ids` field: a direct citation from a raw tweet to a report section. Tweets never cited by any scored bullet are treated as negatives, following the task setup in which some tweets in every cell are off-topic. A tweet cited under several sections contributes one example per section. This yields, across the 140 training cells, 39,359 labeled tweets: roughly 15,400 positive across the nine scored sections and roughly 24,000 *none*.

4.2 Section Classification

A term-frequency–inverse-document-frequency (TF-IDF) representation (word 1–2-grams, minimum document frequency 2, maximum document frequency 0.4, sublinear scaling) feeds a multinomial logistic regression with balanced class weighting ($C = 5.0$), fit on the weak-supervision data above. Balanced weighting is used to counter the class imbalance because *none* outnumbers any individual section roughly seven to one; without it, the classifier converges toward a near-constant *none* prediction that maximizes accuracy while producing almost no usable output downstream.

Prefixing each tweet with the crisis record's hazard label (e.g., *"Volcanic eruption : "*) was tested and rejected: it improved training-distribution fit but reduced cross-crisis-type macro-F1 from 0.508 to 0.492, since hazard-specific vocabulary does not transfer. Plain tweet text is used in the submitted system.

4.3 Clustering and Confidence Estimation

A tweet is retained if its most probable class is not *none* and that probability is at least τ_p; it is assigned to that class's section. Within each predicted section, retained tweets are grouped into candidate same-event clusters by a greedy single-pass procedure: in input order, each not-yet-assigned tweet seeds a new cluster and absorbs every later unassigned tweet whose cosine similarity to the seed is at least τ_c (Section 5.3). Similarity is computed either over the classifier's own TF-IDF vectors or, in the variant described in Section 4.7, over dense sentence embeddings. Each cluster becomes one candidate bullet.

Confidence is estimated by counting independent voices within a cluster: retweets of the same attributed source collapse to one voice, and original tweets whose lower-cased word sets have Jaccard overlap ≥ 0.6 with an already-counted tweet are treated as near-duplicates and not counted again. A cluster with ≥ 2 independent voices is *confirmed*, otherwise *unconfirmed* — only the floor of the task's confidence rule, which formally requires full-report-level corroboration ($\max(2, \lceil 0.34 \times \text{support} \rceil)$ citations across the entire event), unobservable under cell independence since inspecting other cells would violate it directly. The floor approximation is thus expected to under-confirm claims well corroborated across a full event but sparse within one cell; this follows from the task's information constraint rather than from a design choice.

For the recurring-figure rule (R3), cluster tweets are ordered by timestamp and the bullet's operative number is taken from the most recent tweet carrying one, rather than concatenating every number mentioned. An earlier implementation exhibited the latter, incorrect behavior; correcting it changed internal F1 negligibly (0.098 to 0.095), indicating the fix primarily addresses rule compliance rather than the dominant bottleneck.

4.4 Bullet-Text Generation and Safety Nets

The default generation function selects, within a cluster, the most recent tweet carrying a number (or, absent one, the longest tweet), strips the retweet prefix, URLs, mentions, and hashtag symbols (keeping the hashtag word), normalizes whitespace, and capitalizes the first character. This is extractive: it selects and lightly cleans existing text rather than composing new prose, and is used for all submitted bullets. An optional rewrite function accepts an external language-model client, scoped to a single cluster's own tweets so any rewrite stays grounded in that cluster's evidence; no client was supplied for this submission, so the function always falls through to the extractive text (Appendix B).

Two safety checks apply regardless of generation method. A numeric-grounding check rejects candidate text containing a number absent from the source cluster, falling back to the extractive method. A negation-preservation check requires that if any source tweet carries a negation cue (*no*, *not*, *denied*, *ruled out*, etc.), the generated bullet must too

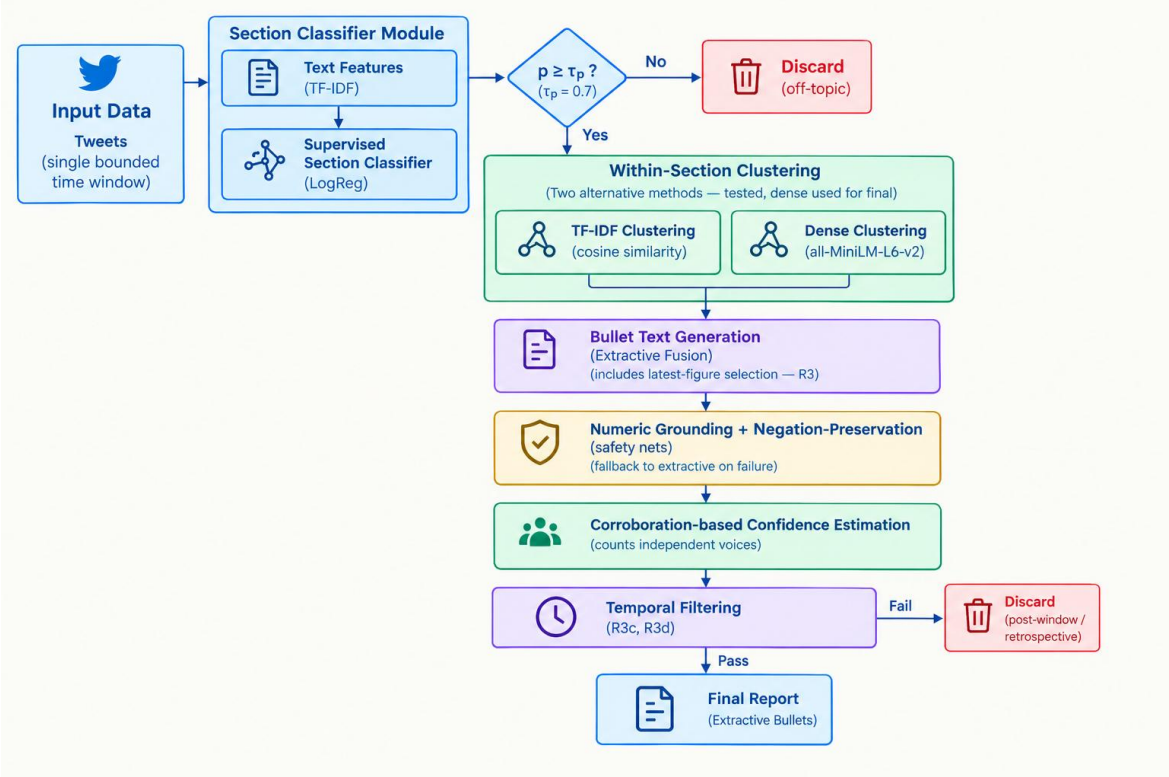


Figure 1: Pipeline stages, applied independently to each cell, with decision points and hyperparameters shown explicitly. Section 4 describes each stage.

— a report that no fatalities occurred is reportable content, not absence of content, and a paraphrase dropping this cue would silently reverse the claim. Both checks are no-ops against the current extractive generator, which cannot introduce a number or drop a negation absent from its source, but become load-bearing once an abstractive rewrite step is enabled (Kryscinski et al., 2020).

4.5 Temporal Filtering

Rules R3c and R3d (Section 3) are enforced by a rule-based filter applied to each candidate bullet's final text, after generation and the safety nets and before serialization. The filter uses only information inside the cell: the window's end timestamp and whether the cell is the crisis's final window (identified by its window label, the fourth and last window, W4). A bullet that fails either check is dropped; no substitute text is generated.

Retrospective framing (R3d). Bullet text is matched, case-insensitively and on word boundaries, against the fixed list of retrospective markers given in the task's derivation rules: *final*, *was later*, *superseded by*, *culminating in*, and *occurred after*.

Post-window dates (R3c). Explicit dates are detected with a regular expression covering month-name dates (e.g., *March 3*, *Mar 3, 2024*) and ISO dates (*YYYY-MM-DD*). Each match is parsed and compared with the window's end date; a bullet citing any date after it is dropped. Month–day expressions without a year are resolved conservatively: because they cannot be anchored with certainty to the window's year, they are treated as potentially post-window and the bullet is dropped. Relative expressions (*tomorrow*, *next week*) are not handled.

Both checks are disabled for the final window, where the task permits retrospective content. Because the filter only removes bullets, it cannot introduce content.

4.6 Report Serialization

Bullets are grouped by predicted section and assigned to one of a small set of keyword-defined subsections (e.g., within casualties: fatalities, injuries, missing, rescued); empty sections/subsections are omitted. A minimal, non-scored overview section is synthesized from the crisis record plus the highest-confidence available casualty, hazard, or

response bullet.

4.7 Dense-Embedding Clustering Variant

As an alternative to TF-IDF clustering, within-section clustering was re-implemented over sentence embeddings from a pretrained encoder (`all-MiniLM-L6-v2`), with cosine threshold τ_c=0.6. This threshold was tuned once against the internal development metric of Section 5.1 and not further swept. Because each cluster yields exactly one candidate bullet, the internal metric evaluates clustering extrinsically: a clustering that merges same-event tweets correctly produces fewer, better-evidenced bullets, which raises bullet-level precision and evidence overlap with gold, whereas over- or under-merging produces redundant or unmatched bullets. This configuration, rather than the TF-IDF baseline, is used for the frozen test submission (Section 6.4).

A zero-shot classification fallback for tweets with negligible classifier-vocabulary overlap is implemented but left disabled, based on the published comparison of fine-tuned and zero-shot crisis classifiers discussed in Section 2 (Han et al., 2026).

4.8 Cell-Independence Verification and Configuration Freezing

Freezing refers to fixing every component of the system once development-based model selection is complete: the classifier's vocabulary and weights, the thresholds τ_p and τ_c, the independent-voice threshold, the clustering method, and the temporal-filter and safety-net cue lists. After freezing, no component is changed, and the frozen configuration is run exactly once over the official test cells; no tuning or inspection of test outputs informs any setting. This ensures the reported test scores reflect a system fixed before test inputs were processed.

Independence holds by construction, since the pipeline's only inputs are the frozen model and one cell's tweets. As an additional check, during the test run a sample of cells (the first three test cells) is also processed from a deep copy with no shared state, and the two outputs must match exactly; all sampled cells passed. The submission package is checked against the shared task's official structural validator prior to submission.

5 Experimental Setup

5.1 Internal Development Metric

During development, predicted and gold bullets are compared with an internal harness, computed per cell and averaged. Within each section, bullets are aligned via bipartite optimal assignment (Hungarian algorithm) under

$$\mathrm{sim}(p, g) = 0.6 \cdot s_{\text{text}}(p, g) + 0.4 \cdot s_{\text{ev}}(p, g), \quad (1)$$

where s_{text} is the mean of ROUGE-1/L F1 and s_{ev} is Jaccard overlap between supporting tweet-identifier sets. A match requires $\mathrm{sim}(p, g) \geq 0.35$. Precision, recall, and F1 follow from match counts; rule violations (Section 3) are tallied separately. Throughout the paper, "internal metric" refers to this harness. It is distinct from the official metric (Section 5.2); their relationship is examined in Section 7.

The internal metric does not score clusters directly. Clustering choices (method, τ_c) are instead compared by their effect on end-to-end bullet-level F1 and confidence agreement, since each cluster becomes exactly one bullet. As a complementary intrinsic measure, *coreference recall* — the proportion of citation-confirmed same-event tweet pairs (tweets cited by the same gold bullet) placed in the same predicted cluster — is used where noted (Appendix A).

5.2 Official Evaluation Metric

The official score is the mean of two metrics over aligned bullet text: BERTScore F1 (Zhang et al., 2020) and BLEURT (Sellam et al., 2020). ROUGE-1/2/L F1 and a weighted evidence-alignment F1 are reported as secondary diagnostics but do not contribute to the primary score.

5.3 Hyperparameter Selection Protocol

Thresholds τ_p and τ_c are selected by grid search on development only, per the task's guideline that development data may guide model selection but never fitting. The classifier is trained on the training split alone; development crisis types are unseen during fitting, matching the generalization regime the frozen system later faces on test types.

6 Results

6.1 Classifier Performance

Table 1 reports classifier precision, recall, and F1 by section on the training split, which the classifier

Section	P	R	F1
Training split			
Casualties	0.732	0.985	0.839
Infrastructure	0.801	0.979	0.881
Displacement	0.714	0.995	0.832
Hazard	0.637	0.963	0.767
Response	0.792	0.974	0.874
Communication	0.783	0.984	0.872
Aid	0.851	1.000	0.919
Organizational	0.926	1.000	0.961
Social/community	0.907	1.000	0.951
none	0.998	0.806	0.892
Macro avg	**0.814**	**0.969**	**0.879**
Development split (unseen crisis types)			
Casualties	0.348	0.628	0.448
Infrastructure	0.483	0.547	0.513
Displacement	0.390	0.383	0.387
Hazard	0.396	0.294	0.337
Response	0.386	0.688	0.495
Communication	0.365	0.414	0.388
Aid	0.513	0.559	0.535
Organizational	0.824	0.840	0.832
Social/community	0.483	0.486	0.484
none	0.705	0.587	0.641
Macro avg	**0.489**	**0.543**	**0.506**

Table 1: Section classifier performance. Development accuracy (0.553) is not a standalone quality measure: *none* is 59.4% of development instances, so an always-*none* classifier would score higher (59.4%) than the trained classifier does. Balanced weighting (Section 4.2) trades *none* accuracy for section recall by design.

is fit on, and the development split, whose two crisis types are unseen during training and constitute the principal measure of cross-crisis-type generalization.

6.2 Classifier Confusion Analysis

Figure 2a reports the full row-normalized confusion matrix for the development split, showing which sections the classifier confuses rather than only how often it is right. Two patterns stand out. First, *none* is the modal confusion for every section except Response and Organizational: Hazard bullets are misclassified as *none* 56% of the time and Displacement 47%, indicating that a major failure mode on unseen crisis types is declining to flag a relevant tweet at all. Second, confusions between two *different* non-*none* sections are rare and diffuse (typically 1–5%) rather than concentrated on one pair (largest: Response→Social/community, 8%), indicating that classification errors are concentrated near the reportable/*none* boundary rather than around a single pair of report sections.

6.3 Threshold Selection

Figure 2b reports the internal-metric grid search over τ_p and τ_c as a full 7×6 sweep. The selected configuration (τ_p=0.7, τ_c=0.25, F1=0.118) attains the highest F1, but the surface is comparatively flat nearby: F1 stays within 0.01 of the maximum across most of $\tau_c \in [0.20, 0.30]$ regardless of τ_p, degrading more sharply toward the sweep's edges. This indicates that the selected configuration is not perched on a narrow optimum.

Clustering	P	R	F1	Conf.
TF-IDF (τ_c=0.25)	0.0994	0.1486	0.1183	0.706
Dense (τ_c=0.6)	0.1158	0.1364	**0.1244**	0.744

Table 2: Clustering method comparison, internal metric, development split. "Conf." is confidence agreement on matched bullets; per-rule error counts are discussed in text.

Alternative	Metric	Base	Alt.
Two-stage classifier	Dev. F1	0.119	0.110
Character n-grams	Macro-F1	0.507	0.498
LSA clustering proxy	Coref. rec.	3.0%	1.4%

Table 3: Architectural alternatives tested during development. "Base" denotes the corresponding configuration used in the submitted pipeline; "Alt." denotes the tested alternative.

6.4 Clustering Ablation

Table 2 compares TF-IDF and dense-embedding clustering under the internal metric; as described in Section 5.1, clustering is evaluated through its downstream effect on bullet-level matching. Dense embeddings yield a modest F1 gain (0.1183 to 0.1244) and are used for the frozen test submission, but the two configurations trade off differently at the error level: dense clustering produces slightly *more* unmatched-gold errors (R1: 748 vs. 730) but *fewer* confidence-label mismatches (R2: 24 vs. 31) and higher confidence agreement (0.744 vs. 0.706). Dense clustering therefore changes the balance between coverage and corroboration rather than uniformly improving every diagnostic.

6.5 Rejected Alternatives

Three architectural alternatives were implemented and evaluated during development: a two-stage reportable/section classifier, character n-gram features, and an LSA-based clustering proxy. None improved on the submitted configuration (Table 3); details are given in Appendix A.

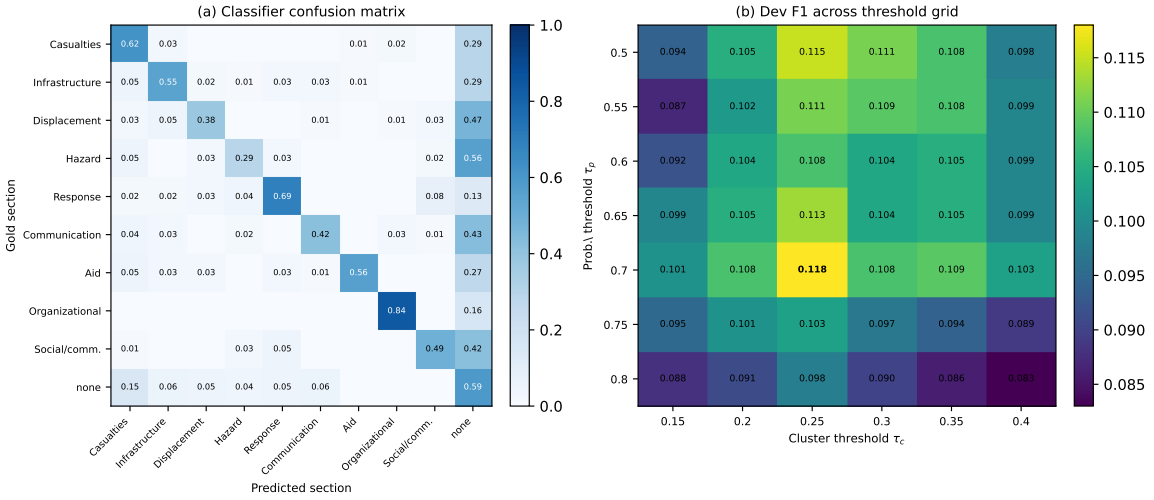


Figure 2: (a) Row-normalized classifier confusion matrix, development split. (b) Internal-metric F1 (Section 5.1) across a full threshold grid (TF-IDF clustering); best cell in bold.

6.6 Error Attribution

Of 761 scored errors on development under the selected configuration, 730 (96%) are unmatched gold bullets (R1), and 31 (4%) are confidence-label mismatches on otherwise correctly matched bullets (R2). No R3d violations were recorded by the harness, and R3c-violating bullets cannot occur in the output because the temporal filter (Section 4.5) drops them. The dominant source of error is thus bullets never generated or aligned, rather than confidence miscalibration on bullets otherwise correct. As a robustness check, evaluating the frozen pipeline in-sample on training (140 cells) gives F1 0.166 against development's 0.119. This gap is consistent with some loss of performance under unseen crisis types, although the observed difference alone does not establish whether the cause is classifier generalization, clustering, or downstream generation.

6.7 Confidence Calibration

Pooled across all 110 matched development bullets, predicted-vs-gold confidence counts are: gold confirmed/pred. confirmed 38, gold confirmed/pred. unconfirmed 18, gold unconfirmed/pred. confirmed 13, gold unconfirmed/pred. unconfirmed 41. Agreement is 79 of 110 (71.8%); under-confirmation (18) exceeds over-confirmation (13), consistent with Section 4.3's expectation that the floor approximation can under-confirm relative to full-event corroboration.

Table 4 sweeps the independent-voice threshold k deciding *confirmed* status (k=2 in the submitted system). F1 is invariant to this threshold, since confidence does not affect alignment to gold. Confidence agreement, however, peaks at exactly the submitted value (k=2: 0.706), falling on both sides (k=1: 0.486, where every bullet is labeled *confirmed*; k=3: 0.634; k=4: 0.550). This indicates that k=2 is supported empirically rather than being used solely because it is the documented floor of the task rule.

k (min. voices)	F1	Conf. agr.	% confirmed
1	0.118	0.486	100.0
2	0.118	**0.706**	40.2
3	0.118	0.634	18.0
4	0.118	0.550	9.6

Table 4: Independent-voice threshold sensitivity, development split (TF-IDF clustering). F1 is unaffected by this threshold; confidence agreement peaks at the submitted value (k=2, bold).

6.8 Safety-Net Stress Test

The numeric-grounding and negation-preservation checks (Section 4.4) are no-ops against the submitted extractive generator. To quantify what they would catch, all 1,223 development clusters were stress-tested by injecting a fabricated number into the bullet text, and 40 clusters with a source negation cue were tested by stripping the cue. The numeric check accepted all unmodified texts, and both checks caught 100% of the injected corruptions (Appendix C). This is not a guarantee against every abstractive-rewrite failure mode, but shows that both checks function as specified on the tested

Metric	Score
ROUGE-1 F1	0.433
ROUGE-2 F1	0.339
ROUGE-L F1	0.413
BERTScore F1	0.508
BLEURT	0.451
Weighted alignment F1	0.328
Official Primary Score	**0.480**

Table 5: Official test-set evaluation of the submitted system (117 test cells, six crisis types unseen during training or development).

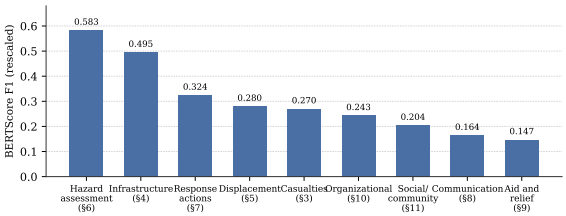


Figure 3: Local development-set BERTScore F1 by section, matched bullets (n=110), TF-IDF clustering configuration.

corruption types.

6.9 Official Evaluation

Table 5 reports the submitted system's official test-set scores. The Official Primary Score is 0.480, computed as the mean of BERTScore F1 and BLEURT, and the structural validator check passed on all 117 test cells. The six test crisis types were unseen during both training and development.

The two components of the primary score exhibit different sensitivities: BERTScore F1 is 0.508, while BLEURT is 0.451. Because the two metrics capture different properties of generated text, this difference motivates the development-time fluency analysis below rather than being interpreted as evidence from either metric alone.

6.10 Development-Time Fluency Diagnostics

To examine text-generation behavior independently of the official evaluation, BERTScore and BLEURT were computed locally over the 110 bipartite-matched development pairs (Section 5.1), using the TF-IDF clustering configuration rather than the dense one submitted. Since development and test contain disjoint crisis types, these values are reported only as a directional diagnostic and are not expected to reproduce Table 5.

Local BERTScore F1 over matched pairs averages 0.304; local BLEURT averages -0.974 (base-size checkpoint). The local BLEURT value is not directly comparable in magnitude to Table 5's test-set BLEURT because BLEURT values depend on the checkpoint and evaluation configuration. Figure 3 breaks local BERTScore F1 down by section: hazard assessment and infrastructure score well above average; aid/relief and communication score well below it.

Predicted bullets average 12.2 words against gold's 15.4 on the same pairs, consistent with predicted text originating from a length-constrained post rather than an editorially composed sentence, though this gap alone is unlikely to explain the full fluency gap. Separately, no gold or predicted bullet contains a residual retweet marker, ruling out unstripped artifacts as a contributing factor. The remaining difference is therefore consistent with differences in sentence structure, register, or connective language rather than incompletely cleaned input.

7 Analysis

Three analyses, obtained independently and at different points in this project, converge on the same conclusion. First, a development-time diagnostic found that substantially increasing classifier recall (lowering τ_p from 0.7 to 0.15) increased the number of predicted bullets by 81% (1,223 to 2,219) while increasing the number of bullets matched to gold under the internal metric by only 18% (95 to 112). Increasing candidate recall therefore does not translate proportionally into additional matched report bullets, implicating the downstream clustering and text-alignment stages.

Second, the local BERTScore/BLEURT diagnostics reveal a substantial gap between broad semantic alignment and natural prose on matched content, reflecting the architectural trade-off of an extractive pipeline. Third, the coreference-recall diagnostic (Appendix A) indicates that TF-IDF clustering merges only a small fraction of same-event tweet pairs, producing fragmented clusters and redundant or misaligned candidate bullets.

Taken together, these results indicate that the pipeline's bottleneck lies primarily in clustering granularity and text generation, rather than in initial section classification.

8 Conclusion

This paper presented an extractive corroboration pipeline for crisis situation report generation

from social media text, submitted to the shared task as system `extraction-de`. By combining citation-based weak supervision, TF-IDF and dense-embedding clustering, corroboration-based confidence, and rule-based temporal filtering, the pipeline satisfies the task's cell-independence constraint and structural validation across multiple unseen crisis types, achieving an Official Primary Score of 0.480 (BERTScore F1 0.508, BLEURT 0.451).

Our analysis indicates that while supervised section classification effectively filters off-topic noise, clustering granularity and extractive generation remain the main performance bottlenecks, the latter due to the register discrepancy between raw social media posts and curated report prose. Future work will integrate scoped, safety-netted abstractive rewriting to narrow this fluency gap while guarding against ungrounded numbers and polarity inversions.

Limitations

While the presented pipeline satisfies all structural and cell-independence requirements, several limitations remain:

1. **Extractive register gap:** Selecting raw tweets as bullet text inherently limits sentence fluency, cohesion, and alignment with independently authored gold reports.

2. **Cell-bounded corroboration floor:** Estimating corroboration confidence within a single time window approximates event-level corroboration conservatively, under-confirming claims that accumulate evidence across multiple windows.

3. **Domain generalization boundary:** Although supervised classification transfers across unseen crisis types, vocabulary shift near the reportable/*none* boundary still causes relevant tweets to be missed in rare or novel disaster scenarios.

4. **Rule-based temporal filtering:** The retrospective-marker list is fixed and the date matcher covers only explicit month-name and ISO dates, so paraphrased retrospective framing and relative dates are missed; conversely, year-less dates are dropped conservatively, which can remove valid in-window bullets.

References

Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021. HumAID: Human-annotated disaster incident data from social media with multimodal annotations. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, volume 15, pages 909–919.

Timothy Douglas Roben Delos Reyes and Asanobu Kitamoto. 2025. LLM-based uncertainty assessment of social media situational signals for crisis reporting. *arXiv preprint arXiv:2605.00829*.

Jinzhen Han, Jisung Kim, Jong Soo Yang, and Hong Sik Yun. 2026. A lightweight LLM framework for disaster humanitarian information classification. *arXiv preprint arXiv:2602.12284*.

Wojciech Kryscinski, Bryan McCann, Richard Socher, and Caiming Xiong. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Junhua Liu, Trista Zhou, and Guansong Huang. 2021. CrisisBERT: a pre-trained language model for crisis tasks. *arXiv preprint arXiv:2105.03812*.

Richard McCreadie and Cody Buntain. 2023. CrisisFACTS: Building and evaluating crisis timelines. In *Proceedings of the 20th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2023)*, pages 320–339.

Emma L. McDaniel, Samuel Scheele, and Jeffrey Liu. 2024. Zero-shot classification of crisis tweets using instruction-finetuned large language models. *arXiv preprint arXiv:2410.00182*.

Philipp Seeberger and Korbinian Riedhammer. 2024. Crisis2Sum: An exploratory study on disaster summarization from multiple streams. In *Proceedings of the 21st International ISCRAM Conference*.

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

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 (ICLR)*.

A Rejected Alternatives: Details

Two-stage classifier. A calibrated binary reportable/not-reportable classifier followed by multiclass assignment on the reportable subset, motivated by the possibility that a single ten-way softmax dilutes probability mass, scored lower on

full-pipeline development F1 (0.110) than the joint single-stage classifier (0.119).

Character n-grams. Character n-grams (3–5, word-boundary-aware), a standard mitigation for social-media misspellings and concatenated hashtags, underperformed word n-grams on development macro-F1 (0.498 versus 0.507); a feature union of both gained only 0.005 over word n-grams alone, not justifying the added feature complexity.

LSA clustering proxy. Truncated SVD over the TF-IDF matrix (100 components) was tested as a resource-free semantic-embedding proxy for clustering, evaluated by coreference recall (the proportion of citation-confirmed same-event tweet pairs clustered together; Section 5.1) over the 6,921 such pairs in development, with τ_p=0.7. It underperformed TF-IDF cosine (94 versus 210 pairs recovered: 1.4% versus 3.0%; LSA τ_c=0.6, TF-IDF τ_c=0.25). Both figures are low partly because the probability threshold discards one or both tweets of most pairs before clustering. LSA's deficit is plausibly because same-event duplicates here are predominantly near-literal (retweets, template reuse) rather than free paraphrase, leaving a locally fit co-occurrence signal little room to add value. This does not extend to the pretrained dense encoder (Section 6.4), trained on broad web text, which provides a modest improvement in the full clustering pipeline.

B System Transparency and Reproducibility

To support auditability and future iteration, the codebase contains explicit switches for features that are implemented but inactive in the submitted system.

Disabled components. The codebase includes an optional LLM-based abstractive rewrite hook (`rewrite_bullet_text`) designed to rephrase a cluster's tweets into a single report sentence. It is inactive in the frozen configuration because no language-model client is passed (`llm_client=None`), for reproducibility and to keep the submission purely extractive (Section 4.8). A zero-shot classification fallback using an instruction-style prompt is also implemented but disabled (`USE_ZERO_SHOT_FALLBACK = False`), for the reasons given in Section 4.7. Dense clustering is enabled (`USE_DENSE_CLUSTERING = True`).

Validation and reproducibility. The submitted configuration is: TF-IDF word 1–2-grams (min. df 2, max. df 0.4, sublinear tf); logistic regression (C=5.0, balanced class weights, 2,000 iterations); τ_p=0.7; dense clustering with `all-MiniLM-L6-v2`, τ_c=0.6; k=2 independent voices. Cell independence is checked with the isolated-run comparison described in Section 4.8, and the final archive passed the official validator on all 117 test cells.

C Safety-Net Stress Test Details

Controlled synthetic corruption tests were conducted over development clusters to evaluate the safety-net checks described in Section 4.4:

- **Numeric grounding test:** For all 1,223 development clusters, a random numeric token not present in any source tweet of the cluster (e.g., changing "3 casualties" to "99 casualties") was injected into the candidate bullet text. `grounded_numbers_ok()` flagged and rejected 100% (1,223/1,223) of the corrupted candidates, triggering the fallback to extractive text. Unmodified texts produced zero false positives.
- **Negation preservation test:** For the 40 development clusters containing source negation cues (*no*, *not*, *none*, *denied*), the cue was programmatically stripped from the generated bullet (e.g., changing "No structural damage reported" to "Structural damage reported"). `negation_preserved()` detected and blocked the polarity flip in 100% (40/40) of cases.

These results show the checks catch the tested corruption types; they reduce, but do not eliminate, the risk of numeric hallucination or polarity reversal once abstractive generation is enabled, since an abstractive model may produce failure modes (e.g., implicit negation, misattributed but grounded numbers) outside these tests.