

# CrisisLink: Closing the Loop between Underspecified Crisis Posts and Humanitarian Coordination

Yuchen Zhang<sup>1,2</sup>, Yuze GAO<sup>1,2</sup>, Bin Chen<sup>1</sup>, Jian Su<sup>5\*</sup>, Farah Benamara<sup>3,4</sup>

<sup>1</sup>A\*STAR Institute of Advanced Intelligence and Computing, Singapore,

<sup>2</sup>CNRS@CREATE LTD, Singapore, <sup>3</sup>IPAL, CNRS-NUS-A\*STAR, Singapore,

<sup>4</sup>IRIT, Université de Toulouse, CNRS, Toulouse INP, Toulouse, France <sup>5</sup>Independent

{zhangyuc, gaoy1, bchen}@a-star.edu.sg, sujian123@gmail.com, farah.benamara@irit.fr

## Abstract

During disasters, social media becomes a critical channel for identifying urgent needs and mobilizing public and institutional response efforts. Prior crisis-informatics research has mainly focused on detecting or extracting actionable content from disaster-related posts. However, real-world humanitarian response requires structured and operationally complete records, while social media posts are often short, informal, and underspecified.

In this work, we present CrisisLink, an LLM-based coordination system that transforms noisy crisis communications into structured crisis records. Unlike prior one-shot extraction systems, CrisisLink dynamically identifies missing operational fields and initiates clarification dialogues with users to repair incomplete records before downstream coordination. The system further grounds resources against a humanitarian taxonomy and supports partial, multi-item matching.

Built on the CrisisNLP (Imran et al., 2016) corpus spanning multiple disaster types, the system is evaluated through human-audited assessments, simulated dispatch workflows, and a pilot evaluation in which trained workers role-play poster replies through a live web-based response interface. Experimental results show that a single clarification round raises end-to-end Offer-Request matching accuracy from 74.0% to 88.3% (a ceiling estimate under full response) while reducing the manual triage burden on responders, demonstrating the potential of interactive LLM systems for real-world humanitarian coordination.

## 1 Introduction

In the wake of disasters, social media posts: requests for aid and offers of help, carry substantial operational value for response efforts, as evidenced

by large-scale corpora such as CrisisNLP (Imran et al., 2016), yet this value is not easily realized in practice. Consider a message posted during a recent flood: "We urgently need water here!!" For a human responder, the intent is unmistakable: someone is asking for help. But for a coordination system that must match needs to supplies, dispatch deliveries, and maintain operational records, the message is unusable, it specifies no location, no quantity, and no recipient. The information needed to act is not in the text, and even a perfect extractor cannot put it there.

A decade of crisis-informatics research has primarily framed this as a problem of extracting more accurately from such posts and matching (Nazer et al., 2016; Basu et al., 2019; Dutt et al., 2019; Basu et al., 2021). We argue that the bottleneck lies elsewhere: the messages themselves are incomplete relative to the schemas downstream systems require, and additional extraction and matching accuracy cannot recover information that was never expressed.

### 1.1 Three Trajectories, One Shared Assumption

Three trajectories of crisis-informatics research converge on a shared assumption that, we argue, is precisely what limits real world deployment.

(a) Extraction and Classification. This line has steadily improved what can be recognized in a crisis post, from cascading SVMs and rule-based detectors (Purohit et al., 2014; Nazer et al., 2016), through neural classifiers (Devaraj et al., 2020; Durham et al., 2023; Dutt et al., 2019), to recent LLM-based pipelines (Ziaullah et al., 2024; Yin et al., 2026; Lamsal et al., 2025; Vitiugin and Purohit, 2024).

(b) Resource matching. A parallel line has paired request-posts with availability-posts: Purohit et al. (Purohit et al., 2014), Basu et al. (Basu et al., 2019, 2021), Dutt et al. (Dutt et al., 2019)

\* Jian Su was with the Institute for Infocomm Research (I2R), A\*STAR, Singapore at the time of this work.

with geographic proximity, and CREMA (Lamsal et al., 2024) with cross-lingual matching.

(c) Actionability. He et al. (He et al., 2017) observed that most signal-bearing crisis posts lack the structured detail required for response; Zade et al. (Zade et al., 2018) formalized this as a gap between situational awareness and actionability; multilingual serviceability ranking (Vitiugin and Purohit, 2024) and topic-based demand modeling (Durham et al., 2023; Zhang et al., 2021) sit in this strand. This line documents the problem but does not act on it: posts deemed non-actionable are filtered out or down-weighted, not repaired.

Across all three trajectories, the crisis post is treated as a fixed artifact: something to recognize from, match on, or filter by, never to edit. Crucially, no prior crisis-informatics work treats a post’s operational incompleteness as something to repair.

## 1.2 From Underspecification to Repair

By operational completeness we mean the following: downstream systems such as procurement, dispatch, and volunteer routing do not consume free text but records, whose required fields are dictated by the operation to be performed. We accordingly distinguish six attributes (United Nations Office for the Coordination of Humanitarian Affairs, 2025; United Nations High Commissioner for Refugees, 2026): time, location, resource type, quantity, urgency, and special requirements.

We adopt this stance. When a disaster is detected, CrisisLink elevates the extraction of requests and offers to top priority. Once a post is identified as a crisis-related request or offer but fails the operational-completeness criterion above, the system uses an LLM to re-examine the fields already extracted, such as the item/location/urgency, and issues targeted clarification questions for those that are missing or insufficiently specific. For instance, in response to a post that simply says "we need plasma", the system would follow up by asking which blood type is needed, and whether medical personnel are available on site to receive it. This recasts the system as an interaction policy, not a one-shot extractor: it must decide when to ask, what to ask, how many rounds to attempt, and when to give up and route the record to manual triage instead. We develop and evaluate this policy in §3.

The general idea of clarification has been developed in task-oriented dialogue with unclear instructions (Gan et al., 2024), ambiguous information seeking (Zhang et al., 2025), and missing-

information-guided retrieval (Siro et al., 2026).

Our setting departs from these in two ways. First, the clarification target is fixed by an external schema, not by user satisfaction or model confidence. Second, users in a crisis frequently do not reply, CrisisLink does not block on the user. When a clarification goes unanswered, the system falls back to a best-effort inference that combines the post’s estimated urgency with the fields already extracted, and proceeds to selective matching and action under that inferred record.

## 1.3 CrisisLink and Contributions

We propose CrisisLink, a system built on DeepSeek-V3.2. Its core objective is accurate Offer-Request matching, pursued through three operations on each target post: clarification, inference, and completion. Around this core, CrisisLink integrates five supporting components: key-field extraction over CrisisNLP; three-level UNSPSC tagging for resource normalization; multi-item matching; and text-to-SQL-backed structured storage. A responder-facing NL query interface is left to future work.

We evaluate CrisisLink at two levels. Offline, three system modules are scored independently: (i) LLM-driven filtering, extraction, and UNSPSC classification, audited by human annotators with inter-annotator agreement reported as Cohen’s  $\kappa$  (McHugh, 2012); (ii) SQL generated from natural-language queries, audited by humans for correctness; and (iii) Offer-Request matching accuracy. Online, we run a small-scale pilot study on a live web interface, in which trained workers role-play poster replies over simulated dispatch cases. Clarifications are presented as one-click options: a single additional click raises the end-to-end matching workflow accuracy from 74.0% to 88.3%. The evaluation design, including the clarification ablation, is reported in §3. This work contributes an end-to-end implementation of CrisisLink, instantiated on the CrisisNLP corpus across nine disaster datasets. In our pilot, a single clarification step improves matching more than scaling extraction or downstream classification alone would, a controlled-setting result, with real-deployment caveats in the Limitations.

## 2 CrisisLink System Architecture

CrisisLink converts crisis-related social media posts into schema-conformant, taxonomy-

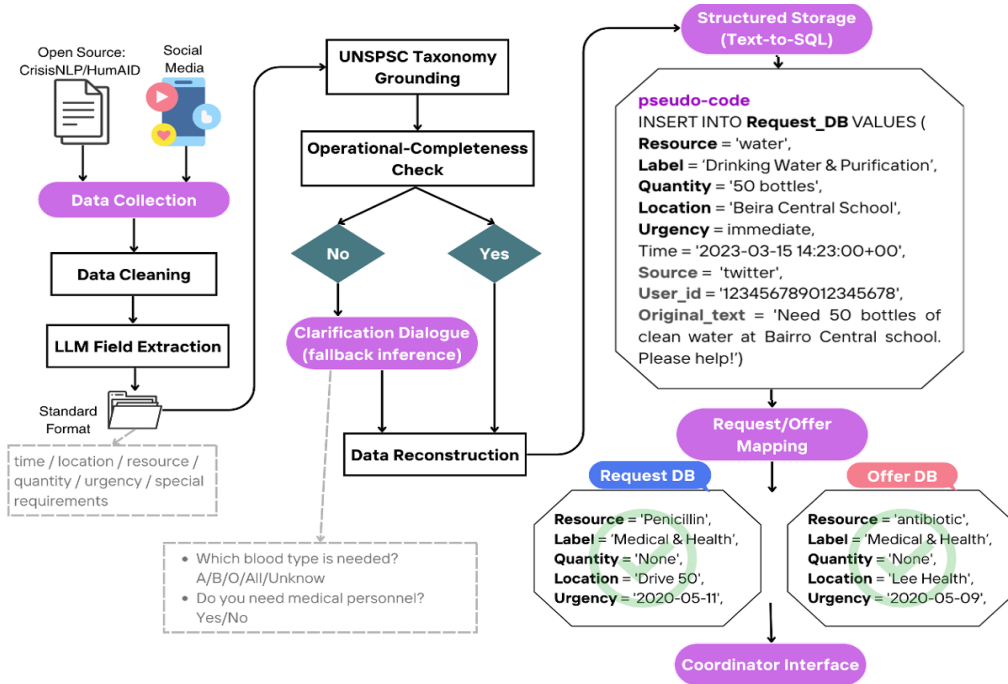


Figure 1: The CrisisLink pipeline

grounded supply/demand records and makes those records available to responders for matching and dispatch. Figure 1 summarizes the pipeline.

Posts are acquired, normalized, and passed through LLM field extraction into a fixed-attribute format (§2.1). Each record then enters a closed-loop routine that grounds it against a curated UNSPSC subset (§2.2), tests it against an operational-completeness criterion, and—when the record fails—clarifies, infers, re-grounds, and reconstructs it (§2.3), re-testing until the record either passes or is routed to manual triage. Completed records are persisted into request and offer stores via LLM-generated, structurally validated SQL (§2.4), and exposed to responders through supply–demand matching and a coordinator interface (§2.5). *The pipeline is closed-loop where information is missing*: when a record fails the completeness criterion, the system actively repairs it through clarification or inference, rather than filtering it out as prior work does.

## 2.1 Ingestion and Field Extraction

CrisisLink ingests posts from open crisis corpora and, in the pilot, a real-time simulation generator, normalizing all sources into a common format. Each post is processed by a single LLM call that emits an extraction record over the 6 attributes schema of: **time, location, resource type, quantity, urgency, special requirements**, and two non-exclusive labels, **request, offer**.

The prompt enforces one explicit design choice, **specificity over coverage**: a post is retained only if it names a concrete resource (“plasma”), not merely a need for “help,” since a request with nothing to match on cannot enter matching. A subsequent disambiguation pass handles generic terms (*supplies, aid*, etc.): from context, the model specifies or deduplicates the term, or keeps it as generic when genuinely uncertain. A kept generic term is not a dead end but an explicit gap, deferred to the completeness check (§2.3). (Appendices A, B)

## 2.2 UNSPSC Taxonomy Grounding

CrisisLink grounds every retained resource against the UNSPSC catalog at three levels (segment / family / class). During curation, each segment/family/class is assigned a static ‘criticality’ score (how life-critical the resource is in humanitarian response), labeled by the LLM and confirmed in the same human review. The repair policy of §2.3 (described next) uses it to prioritize clarification and to escalate unanswered high-criticality records. Grounding sits *inside* the repair loop: because a clarification round can revise the resource itself, like turning a generic “food” into a specific “infant formula”, a reconstructed record is re-grounded before it is re-checked.

Mapping against the full UNSPSC catalog (149,846 classes) is inefficient and noisy, as large segments such as industrial machinery have no role in humanitarian aid; we therefore curate a relevant

subset once(58 segment/ 658 family/ 8,637 class), via an LLM relevance pass confirmed by a human reviewer (§3 reports coverage). Item mapping then proceeds from human-seeded high-frequency items to fully automatic classification: seeding the high-frequency head curbs repeated LLM calls on the items that recur most often. (Appendix C)

### 2.3 Closed-Loop Completeness Check, Clarification, and Reconstruction

**Operational completeness.** A grounded record is tested against the six-attribute schema of §2.1. Each attribute is classified as *present* (a value populated upstream that survives a specificity check), *inferable* (recoverable from the post itself, the poster’s own earlier posts in the same thread, or the dataset-level disaster context, with confidence above threshold), or *missing*. A record passes when all critical attributes are present or inferable above threshold; otherwise its non-empty *gap set* enters the repair loop. This check is a *gate*, not a filter: failing it does not discard the record, it routes the record to repair.

**The repair loop.** For each record with a non-empty gap set, CrisisLink runs three operations and re-tests the result:

- **Clarification.** The system generates a follow-up to the poster, governed by two principles. *What to ask:* when several attributes are missing, the system ranks the gap set by operational priority, and this ranking is *category-dependent*. For broad-relief categories such as food, water, and shelter, quantity and location dominate, how many people need it and where matters more than finer resource detail. For medical needs the priority inverts: the specific resource (blood plasma vs. saline vs. insulin) is more critical. The system asks the highest-priority gap for the record’s category first, so that a single answer resolves, as far as possible, the most consequential gap; the category-to-priority mapping is supplied to the LLM as part of the clarification prompt (Appendix A). *How to ask:* when a gap admits a small fixed answer set, the question is posed as a *one-click multiple choice* (“which blood type is needed?”) ∈ {A, B, O, All, Unknown}); free text is used only otherwise. A single click is meant to resolve the most consequential gap with the least cost to the poster. The dialogue admits at most three rounds per post.

- **Fallback inference.** If clarification does not return within the configured window, CrisisLink does not block. The inference module conditions on the post text, the extracted fields, the dataset-level disaster context, and the poster’s own earlier posts in the same thread. Incorporating posts from *other* nearby users is a natural extension we leave to future work. Consider a post from the 2015 Nepal earthquake corpus: “*Mom is still trapped under the rubble, we need rescue equipment now please!!*” Clarification is issued but does not return. Inference then draws on event metadata and the poster’s earlier thread post (“our building on Sankhamul road”) to recover **location**, and on in-text cues to set **urgency**=immediate. Inferred values carry a confidence score and are never merged with *present* ones. Crucially, inference fills only context attributes and leaves the generic *rescue equipment* unchanged—it never commits to “hydraulic cutters” the post never named; unrecoverable attributes such as **quantity** stay explicitly missing.

- **Reconstruction.** The clarified or inferred values are written back into the record. Because a clarification can revise the resource itself—most often by resolving a generic term into a specific one—the reconstructed record is re-grounded against UNSPSC (§2.2) and re-submitted to the completeness check. A record that now passes proceeds to storage; one that still fails after the round budget is exhausted is, when its urgency is high, routed to a *manual triage* queue with its unresolved attributes flagged, rather than emitted silently into matching.

**Policy.** Treating repair as an interaction policy introduces two decisions with no analog in single-pass systems. **When to ask:** clarification is suppressed on posts already estimated as IMMEDIATE urgency, unless the missing attribute is itself required for immediate dispatch (e.g., the post carries no geo-coordinates), so as not to add interaction cost on a user who needs action rather than questions. **When to stop:** the system backs off on posters who have ignored prior clarifications, and counts re-grounding and re-checking within the same three-round budget, so the loop is bounded by construction.

**Walkthrough.** As a concrete illustration, consider a post from the 2015 Nepal earthquake sub-corpus: “Severe burns patient at Patan Hospital, need plasma immediately, please help.” Extraction yields request=true, resource=plasma, location=Patan Hospital, urgency=immediate. Plasma grounds to a UNSPSC family, however, by the criterion of §2.3, a medical resource grounded only to a **segment** or **family**, rather than to a leaf node **class**, is still not specific enough, and the system must determine which specific resource is meant.

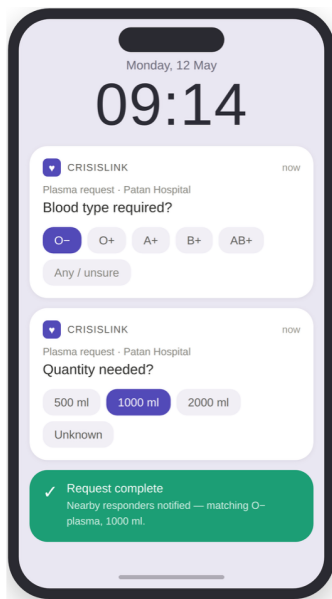


Figure 2: One-click clarification workflow in CrisisLink.

CrisisLink therefore issues a two-question clarification (Figure 2); a reply of *O-negative* and *1000 ml* closes both gaps, the record passes the re-check, and matching is restricted to inventory of O-negative, 1000 ml plasma. Had the poster not replied, the blood type could not be inferred from any context signal, and the record would, given its ‘urgency’ and ‘criticality’ score, be routed to manual triage.

## 2.4 Persistence and Querying

Records that pass the completeness check are inserted into relational request and offer database, keyed on (post\_id, item) (Figure 1 shows an insert example) so that re-ingesting a record updates it in place rather than creating a duplicate. The schema is fixed, not generated per run: it must stay stable for downstream matching and query stages. CrisisLink therefore uses text-to-SQL sparingly: the LLM generates the parameterized INSERT statement once per schema version, not once per record.

Each generated statement is validated structurally (statement type, target table, placeholder count) before use.

A validated statement is cached and reused for all subsequent writes; on a human-audited sample of 1,000 inserted records (Appendix C), the stored fields match the source post at 93% accuracy (§3). The same machinery can run in reverse: a natural-language responder query (e.g., “unmet medical needs in District 3”) would be translated into a validated SELECT over the same schema. A full responder-facing query interface is left to future work.

## 2.5 Supply–Demand Matching and Responder Interface

The final stage turns records back into operational action. Matching operates over the persisted request and offer stores in three modes, applied in priority order (Figure 3):

- **Exact match:** a REQUEST and an OFFER share an item and, after location normalization, a serviceable area. Quantity is reconciled by the matcher rather than treated as a hard constraint, since partial fulfilment is useful in crisis dispatch.
- **Sibling-class match:** when no exact match exists, the matcher admits offers whose UNSPSC class shares a family with the need’s class, ranked by a substitution penalty derived from the taxonomy structure. This is the operational reason for grounding items in a hierarchy rather than a flat vocabulary (§2.2).
- **Multi-item match:** a single REQUEST carrying  $k$  items is matched against a set of offers that *jointly* cover it. A request for {insulin, syringes, alcohol swabs} against an offer pool holding only insulin yields an accepted partial match plus a residual queue for the remaining items, rather than rejecting the whole request.

Matched candidates are surfaced to responders through a web interface, each with the full provenance chain, original post, extracted record, any clarification exchanges, taxonomy path, and matching mode, so that every proposed match is auditable. A responder either accepts a match, which writes a dispatch record back to the store, or rejects it with a structured reason.

**NEED** Beira Central School · urgency: high  
Requested: tents ×20, blankets, drinking water  
"Help, we are trapped at Beira Central School, we need tents, blankets and water."

3 candidate matches, ranked

**EXACT MATCH** Offer #0876 · Red Cross depot covers: drinking water  
Quantity: 200 bottles  
Location: Marginal Road  
provenance: post · record · taxonomy · mode Accept Reject

**SIBLING-CLASS** Offer #0913 · local supplier covers: tents  
Same family as tent: tarpaulins. Substitution penalty: low.  
Location: Lighthouse Avenue  
provenance: post · record · taxonomy · mode Accept Reject

**RESIDUAL** blankets — no offer yet held in queue

Figure 3: Simulation example of Offer-Request matching

### 3 Evaluation and Analysis

Because CrisisLink produces records under a schema prior crisis-informatics systems do not emit, no off-the-shelf baseline operates on comparable outputs. We therefore evaluate each component through a human acceptance audit. For each sampled record, CrisisLink first produces an output, which two annotators then independently assess for correctness under task-specific guidelines (Appendix C); their judgments are reconciled through discussion into a consensus label, against which the reported accuracy is computed. We audit 1,000 records per offline component and 1,500 Offer-Request pairs (500 of them simulation pairs) for matching.

For matching, we report Human-judged **Operational Match Accuracy**, defined as (proposed matches judged operationally valid) / (evaluated proposed matches), where a (REQUEST, OFFER) pair is valid if the offer satisfies the request’s operational constraints; which constraints bind is resource-dependent (type, quantity, location, urgency). Annotators judge the pair the system returns rather than agreement with a pre-selected gold pair, so an invalid return counts as a failed match. The 74.0% and 88.3% conditions below are evaluated on the same 1,500 pairs, which is what licenses the paired McNemar test. Inter-annotator agreement is reported as Cohen’s  $\kappa$ , measured on the independent judgments prior to reconciliation. These figures characterize the output quality of emitted records, the quantity that matters once a post enters the pipeline. Coverage over the raw stream is governed by the upstream specificity-over-coverage filter (§2.1); we report the resulting funnel next.

**Pipeline funnel.** On the donation-related CrisisNLP/HumAID subset, 64% of ingested posts pass the specificity gate (§2.1); the remaining 36% are situational-awareness, emotional, or under-specific posts. Of retained posts, 59% are complete on ingestion and enter matching directly; the other 41% need clarification. Of those, 53% are resolved by clarification, 36% by fallback inference, and 11% are routed to manual triage (about 4.5% of all retained posts). Triage fires only when both clarification and inference fail, or when a high-criticality request remains incomplete. Raw-stream *recall* (the share of genuinely actionable posts the system captures), would require an exhaustively labeled ground-truth stream we do not have, and is left to future work.

| Task                                           | Acc. (%)    | $\kappa$ |
|------------------------------------------------|-------------|----------|
| Offer/Request Classification                   | 87.1        | 0.88     |
| Item Extraction                                | 79.4        | 0.84     |
| UNSPSC Grounding                               | 81.6        | 0.72     |
| Ingestion Post→SQL Correctness                 | 93.2        | 0.89     |
| Matching w/o Clarification<br>(inference only) | 74.0        | 0.76     |
| Matching w/ One Clarification                  | <b>88.3</b> | 0.81     |

Table 1: Component-wise human acceptance accuracy on emitted outputs.

Among the offline components, item extraction is the noisiest at 79.4%, dominated by posts that name a resource only implicitly (“*The tap has been dry for two days.*”) and by extractor confusion between population head-counts and resource quantities; UNSPSC grounding at 81.6% carries part of this error downstream, since a misplaced class constrains both the persisted segment fields and the sibling-class matching pool (§2.5). The NL→SQL component is the strongest at 93.2%, consistent with the schema being fixed across runs and the LLM being invoked once per schema version rather than once per record.

The central result of this work is in the last two rows of Table 1: a single clarification round lifts the match accuracy from 74.0% to 88.3% on 1500 Offer-Request pairs, a 14.3-point absolute gain (paired McNemar,  $p < 0.001$ ) at a median cost of one click per poster. **To our knowledge, CrisisLink is among the first crisis-informatics systems to incorporate an explicit clarification loop that solicits missing information directly from users during the Offer-Request matching process.** In our pilot, this loop converts most otherwise-unmatchable underspecified records into actionable matches.

**Clarification ablation.** We isolate the contribution of gap selection (Table 2). The two clarification conditions share the backbone, the prompts, and the interaction budget, and differ only in which gap the question targets. The fixed-priority baseline always asks the two most frequently missing fields, location and quantity, the strongest fixed ordering available, and reaches 79.2%, against 88.3% for our category-dependent policy. The 9.1-point difference shows that the gain comes from asking about the *most informative* missing field, not from the act of asking.

| Gap-selection policy                       | Match Acc. (%) |
|--------------------------------------------|----------------|
| No clarification (inference only)          | 74.0           |
| Fixed global priority (location, quantity) | 79.2           |
| Category-dependent (ours)                  | <b>88.3</b>    |

Table 2: Clarification ablation on the same 1,500 pairs. All three rows share the backbone, prompts, and interaction budget (one round,  $\leq 2$  one-click questions); only the choice of *which* gap to ask about differs.

**Reading 88.3% as a ceiling.** We report a single clarification round throughout, although the system permits three, because repeated interaction cannot be assumed of a poster in crisis. Even so, clarification replies in the pilot come from trained role-players, so every case that is asked receives an answer. The two reported numbers therefore bracket an operating range rather than predict deployment performance: 74.0% is a floor requiring no user response at all, and 88.3% assumes a reply whenever one is requested. Writing  $r$  for the rate of *useful* replies, linear interpolation gives  $\text{Acc}(r) \approx 74.0 + 14.3r$ , e.g. 81.2% at  $r=0.5$ . This is an optimistic bound: it treats whether a poster replies as independent of how much that reply would help, whereas the hardest cases, posters who do not know the answer are also the least likely to reply. Real deployment should fall below this line but above the 74.0% floor. The caveat applies only to the clarification lift; the offline accuracies in Table 1 are measured on real corpus posts.

Qualitative coding of failed matches surfaces three recurring patterns: rhetorical posts that pass REQUEST classification but name no recoverable resource (“*we need a miracle*”); brand-name medications outside the curated UNSPSC class set, which currently degrade to family-level grounding rather than triggering a user-additions extension; and multi-event threads, where inference cannot yet tell apart posts belonging to different events in

the same thread.

## 4 Conclusion

We presented CRISISLINK, an LLM-based coordination system that converts noisy disaster social media into UNSPSC-grounded, schema-conformant supply-demand records and exposes them to responders for matching. The central design move is to stop treating crisis posts as fixed artifacts: when a record fails the operational-completeness criterion, the system clarifies the gap with the poster, falls back to bounded inference when no reply arrives, re-grounds the reconstructed record, and re-tests it, rather than discarding or carrying it forward incomplete. On a CrisisNLP-based audit, this closed-loop design lifts match accuracy by 14.3 points at a median cost of one click per poster, suggesting that the next consequential gain in crisis-informatics lies less in larger extraction models than in interaction policies that repair the information a post never carried.

## Limitations

**Pilot vs. real-time deployment.** The matching audit was conducted on Offer-Request pairs replayed through the pilot interface, with clarification responses produced by trained workers reading persona cards rather than by the original posters in an active crisis. Both the response rate and the 14.3-point lift to 88.3% should be read as ceiling estimates: a real-time deployment would face lower engagement (likely dominant), content-moderation noise, deletions, and adversarial content, none of which the present pipeline handles. Quantifying end-to-end coverage over the raw stream, including the implicit-resource cases of Appendix G, is left to future work.

**Single backbone.** All LLM components run on DeepSeek-V3.2. Because both the 74.0% and 88.3% conditions share this backbone, the 14.3-point lift isolates the clarification mechanism under a fixed model; the absolute figures may not transfer to other models, and a multi-backbone study is left to future work.

**Language coverage.** CrisisNLP is dominated by English, with shorter Spanish, Tagalog, and Hindi tails on which item extraction degrades. The curated UNSPSC subset was reviewed only against English class labels, and CrisisLink does not currently generate clarification prompts in the poster’s

inferred language. The extraction stage already runs on a multilingual backbone, so the missing piece is multilingual prompt design and evaluation on non-English sub-corpora, not a model retraining.

**Clinical-grade substitution scores.** The substitution scores used by sibling-class matching come from an LLM-prompt rule set rather than medical adjudication; the hard-block rules (e.g., rapid-acting vs. long-acting insulin, blood-type incompatibility) are conservative defaults, not a clinically validated registry, and we leave it to future work.

## Ethics Statement

**Data.** All posts come from CrisisNLP and HumAID, publicly released research corpora used under their stated licences. We collect no additional data from any platform, store no user identifiers beyond the corpus-provided post IDs, and did not attempt to re-identify any poster. Fields that could be sensitive when populated (`location_geo`, `special_req`) exist in the schema for the operational use case but, in this study, were populated only from corpus text and held on access-controlled institutional storage.

**Participants and approval.** No crisis-affected person was contacted at any point. Clarification replies in the pilot were produced by three volunteer lab members role-playing persona cards; outputs were audited by two lab members, with a third resolving deadlocks. All were adults who gave informed consent, participated without compensation, and had no personal data recorded.

**Risk of harm from inference and substitution.** An incorrect inferred location or an unsafe resource substitution could misdirect aid. Safeguards *implemented* in the audited system: inference never specializes a generic resource, never invents one the post did not name, and never infers patient-specific medical data such as blood type (App. A.6); inferred values carry provenance and a confidence floor of 0.70 and are never merged with stated values; every proposed match carries its full provenance chain; and no match is dispatched without a responder’s explicit acceptance. Blood-product and medication matches in particular are advisory only and require confirmation by qualified personnel. Safeguards that remain *unimplemented*: clinical validation of the substitution registry, poster opt-in, platform-level coordination, and handling of

adversarial or moderated content. These are prerequisites for deployment, not features of the present pilot.

**Interacting with people in crisis.** Initiating a clarification dialogue with a poster in an active emergency raises questions of consent, identity, and cognitive load. The design choices reported here, one round by default, one-click options, suppression of questions on posts already judged immediate, and per-poster back-off, are intended to minimize that burden, but they do not substitute for oversight by a humanitarian operations partner, which any real deployment would require.

## Acknowledgments

This research is partially supported by the programme DesCartes funded by the National Research Foundation, Prime Minister’s Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

## References

- Moumita Basu, Sipra Dasbit, and Saptarshi Ghosh. 2021. [Utilizing microblogs for optimized real-time resource allocation in post-disaster scenarios](#). *Social Network Analysis and Mining*, 12.
- Moumita Basu, Anurag Shandilya, Prannay Khosla, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. [Extracting resource needs and availabilities from microblogs for aiding post-disaster relief operations](#). *IEEE Transactions on Computational Social Systems*, 6(3):604–618.
- Ashwin Devaraj, Dhiraj Murthy, and Aman Dontula. 2020. [Machine-learning methods for identifying social media-based requests for urgent help during hurricanes](#). *International Journal of Disaster Risk Reduction*, 51:101757.
- James Durham, Sudipta Chowdhury, and Ammar Alzarrad. 2023. [Unveiling key themes and establishing a hierarchical taxonomy of disaster-related tweets: A text mining approach for enhanced emergency management planning](#). *Information*, 14(7).
- Ritam Dutt, Moumita Basu, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. [Utilizing microblogs for assisting post-disaster relief operations via matching resource needs and availabilities](#). *Information Processing Management*, 56(5):1680–1697.
- Yujian Gan, Changling Li, Jinxia Xie, Luou Wen, Matthew Purver, and Massimo Poesio. 2024. [Clarq-llm: A benchmark for models clarifying and requesting information in task-oriented dialog](#). *ArXiv*, abs/2409.06097.

- Xingsheng He, Di Lu, Drew Margolin, Mengdi Wang, Salma El Idrissi, and Yu-Ru Lin. 2017. [The signals and noise: Actionable information in improvised social media channels during a disaster](#). In *Proceedings of the 2017 ACM on Web Science Conference, WebSci '17*, page 33–42, New York, NY, USA. Association for Computing Machinery.
- Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. [Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages](#). In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 1638–1643.
- Rabindra Lamsal, Maria Rodriguez Read, Shanika Karunasekera, and Muhammad Imran. 2024. [Crema: Crisis response through computational identification and matching of cross-lingual requests and offers shared on social media](#). *Preprint*, arXiv:2405.11897.
- Rabindra Lamsal, Maria Rodriguez Read, Shanika Karunasekera, and Muhammad Imran. 2025. ["actionable help" in crises: A novel dataset and resource-efficient models for identifying request and offer social media posts](#). *ArXiv*, abs/2502.16839.
- Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3):276–282.
- Tahora H. Nazer, Fred Morstatter, Harsh Dani, and Huan Liu. 2016. [Finding requests in social media for disaster relief](#). In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1410–1413.
- Hemant Purohit, Carlos Castillo, Fernando Diaz, Amit Sheth, and Patrick Meier. 2014. [Emergency-relief coordination on social media: Automatically matching resource requests and offers](#). *First Monday*, 19.
- Clemencia Siro, Yifei Yuan, Mohammad Aliannejadi, and Maarten de Rijke. 2026. [Agent-cq: Automatic generation and evaluation of clarifying questions for conversational search with large language models](#). *ACM Trans. Inf. Syst.*, 44(5).
- United Nations High Commissioner for Refugees. 2026. [Needs assessment for refugee emergencies \(NARE\)](#). UNHCR Emergency Handbook. Last updated 11 March 2026.
- United Nations Office for the Coordination of Humanitarian Affairs. 2025. [Global humanitarian overview 2026](#). Technical report, OCHA, New York. Launched 8 December 2025.
- Fedor Vitiugin and Hemant Purohit. 2024. [Multilingual serviceability model for detecting and ranking help requests on social media during disasters](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):1571–1584.
- Kai Yin, Bo Li, Chengkai Liu, Ali Mostafavi, and Xia Hu. 2026. [Crisissense-llm: instruction fine-tuned large language model for multi-label social media text classification in disaster informatics](#). *Advanced Engineering Informatics*, 75:104836.
- Himanshu Zade, Kushal Shah, Vaibhavi Rangarajan, Priyanka Kshirsagar, Muhammad Imran, and Kate Starbird. 2018. [From situational awareness to actionability: Towards improving the utility of social media data for crisis response](#). *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW).
- Michael JQ Zhang, W. Bradley Knox, and Eunsol Choi. 2025. [Modeling future conversation turns to teach LLMs to ask clarifying questions](#). In *The Thirteenth International Conference on Learning Representations*.
- Ting Zhang, Shi Shen, Changxiu Cheng, Kai Su, and Xiangxue Zhang. 2021. [A topic model based framework for identifying the distribution of demand for relief supplies using social media data](#). *International Journal of Geographical Information Science*, 35(11):2216–2237.
- Abdul Wahab Ziaullah, Muhammad Imran, and Ferda Ofli. 2024. [Monitoring critical infrastructure facilities during disasters using large language models](#). *Proceedings of the International ISCRAM Conference*, 21.

## A Prompt Templates

We reproduce here the system and task prompts used by each LLM call in CrisisLink. All prompts are rendered to DeepSeek-V3.2 in JSON mode; outputs are parsed against the Pydantic schemas of Appendix B, with up to three structural retries on parse failure.

### A.1 Stage 2 — Field Extraction (§2.1)

SYSTEM

You are a humanitarian classification assistant for crisis social media. You read one short post at a time and produce a structured JSON record. Be conservative; do not infer beyond the post.

USER

POST: {post\_text}

EVENT\_CONTEXT: {event\_name}, {event\_phase}

Return a JSON object with the following fields:

```

request      : true | false
offer        : true | false
request_conf : float in [0, 1]
offer_conf   : float in [0, 1]
resource_type: str -- specific noun phrase
               naming the resource, or ""
quantity     : str -- "<number> <unit>", or ""
location     : str -- verbatim place mention
time         : str -- post-internal time anchor
urgency      : "immediate" | "high"
               | "moderate" | "low" | "unknown"
special_req  : list[str] -- e.g., blood type,
               age stage, formulation

```

Rules:

- R1. request = true only if the post explicitly requests aid for a concrete, named resource. "We need help" alone is request=false.
- R2. offer = true only if the post explicitly provides or commits a concrete, named resource.
- R3. resource\_type must be a short noun phrase ("plasma", "infant formula"), not a sentence.
- R4. Generic terms (supplies, aid, equipment, materials, resources) are admitted only if no specific noun phrase is available.
- R5. Output ONLY the JSON object.

## A.2 Stage 2 — Generic-term Disambiguation

SYSTEM

You disambiguate generic resource terms in crisis posts. You are CONSERVATIVE: if context does not unambiguously fix the referent, you preserve the generic and flag it for downstream clarification.

USER

POST: {post\_text}

ITEMS (generics in bold): {item\_list}

For each generic in {supplies, aid, equipment, materials, resources}, choose ONE action:

- (A) "replace" -- context UNAMBIGUOUSLY names the referent. Provide specific replacement(s).
- (B) "drop" -- generic is a synonym of another item already in the list.
- (C) "keep" -- genuinely underspecified. Preserve the generic and flag for clarification.

When in doubt, choose (C). Do not invent specific items not warranted by the post.

Return JSON:

```
{
  "items_resolved": list[str],
  "items_needs_clarification": list[str],
  "reasoning": str
}
```

## A.3 UNSPSC Grounding (§2.2)

SYSTEM

You map a crisis-relief resource to the curated UNSPSC subset. You propose 1-3 candidate paths at DIFFERENT DEPTHS so that the system can choose a granularity that downstream matching can support. You do NOT invent class names.

USER

RESOURCE: {resource}

EXAMPLES: {up\_to\_3\_example\_posts}

TAXONOMY: {curated\_segment\_family\_class\_listing}

Constraints:

- C1. Candidate depth must vary. Provide at least one segment-only (depth=1) and one segment+family (depth=2). Add a segment+family+class (depth=3) only if a curated class clearly matches.
- C2. Brand names map to their generic class ("Lantus" -> insulin glargine class).
- C3. If no curated class matches, return the depth-2 fallback as the recommended candidate.

Return JSON:

```
{
  "recommended_index": int,
  "candidates": [
    {
      "depth": 1|2|3,
      "segment": str,
      "family": str|null,
      "class": str|null,
      "confidence": float,
      "rationale": str,
    }, ...
  ]
}
```

## A.4 Completeness Check (§2.3)

SYSTEM

You are an operational-completeness checker. Given an extracted crisis record and the six-attribute schema, you classify each attribute as PRESENT, INFERABLE, or MISSING, and you flag resources that are nominally specific but operationally insufficient (e.g., "plasma" without ABO/Rh type).

USER

RECORD: {record\_json}

EVENT\_CONTEXT: {event\_name}, {event\_phase}

NEARBY\_POSTS: {nearby\_posts\_or\_none}

For each of {time, location, resource\_type, quantity, urgency, special\_req}, return:  
"present" | "inferable" | "missing"

Additionally, mark the resource as "underspecified\_for\_match" and list sub-attributes required for dispatch, using the registry below:

```
plasma / whole blood :
                        {ABO/Rh, recipient capability}
insulin                : {formulation, strength}
oxygen                 : {delivery mode, flow rate}
antivenom              : {species}
infant formula         : {age stage}
```

Return JSON:

```
{
  "attributes": { ... per-attribute status ... },
  "underspecified_for_match": list[str],
  "gaps_blocking_match": list[str]
}
```

## A.5 Clarification Question Generation

The clarification prompt is supplied with the category-to-priority mapping of Appendix E.

SYSTEM

You compose at most two clarification questions for a crisis poster. Each question targets ONE gap attribute, is presented as a one-click multiple choice when the gap admits a small enumeration, and free text only otherwise. Assume the poster is under stress; use plain language and no preamble.

USER

RECORD: {record\_json}

GAPS: {gaps\_blocking\_match}

UNDERSPECIFIED: {underspecified\_for\_match}

RESOURCE\_CATEGORY: {category} -- food, water, shelter, medical, logistics, etc.

PRIORITY\_MAP: {category\_to\_priority\_table}  
-- see Appendix E

Option sets MUST be derived from the inferred context, not from a fixed global list. The LAST

option must be a graceful exit ("other", "don't know", "any compatible").

Return JSON:

```
{
  "questions": [
    {"gap_attribute": str,
     "prompt": str,
     "option_set": list[str]}, ...
  ],
  "expected_information_gain": list[float]
}
```

## A.6 Fallback Inference

SYSTEM

You fill missing CONTEXT attributes in a crisis record. You must NEVER:

- specialize a generic resource into a specific one;
- invent a resource the post does not name;
- infer patient-specific medical data (blood type, prescription) from external signals.

You MAY infer location, urgency, time, and approximate quantity when post or thread cues support a confidence  $\geq 0.70$ .

USER

```
RECORD:      {record_json}
GAPS:        {missing_attributes}
EVENT_CONTEXT: {event_name}, {event_phase}
NEARBY_POSTS: {nearby_posts_or_none}
POST_TIMESTAMP: {ts}
```

Return JSON:

```
{
  "inferred": { "<attr>":
    {"value": str, "confidence": float,
     "provenance": str,
     "source": "inferred"|"inferred_low_conf"}, ...},
  "not_inferable": list[str]
}
```

In the audited runs, NEARBY\_POSTS is always none; cross-post inference is left to future work.

## A.7 Sibling-class Substitution Scoring (§2.5)

SYSTEM

You score whether one crisis-relief resource can operationally substitute for another in dispatch, on [0, 1]. You apply clinical and functional rules. You do NOT consider quantity, location, or urgency.

USER

```
REQUESTED: {requested}
CANDIDATES (same family): {candidate_list}
```

For each candidate, return:

- score in [0, 1]
- hard\_block: true if substitution is unsafe regardless of other factors
- rationale: one sentence

Hard-block examples (not exhaustive):

- rapid-acting  $\leftrightarrow$  long-acting insulin
- 0+ blood  $\rightarrow$  0- patient (one-way only)
- cow milk formula  $\rightarrow$  infant under 6 months

Return JSON: list[{candidate, score, hard\_block, rationale}]

## A.8 Text-to-SQL — Ingestion INSERT

SYSTEM

You produce a single SQLite INSERT that binds the columns of the request\_records (or offer\_records) table to ? placeholders. You never inline literals; every value is a ? placeholder.

USER

TARGET SCHEMA: {full\_DDL} -- see Appendix B.2

Required:

- Exactly one INSERT statement (no SELECT, DDL).
- Target table MUST be one of the registered names.
- Exactly K placeholders, K=column count minus id.
- Use INSERT OR REPLACE to honor UNIQUE (post\_id, item).
- Wrap output in ``sql ... `` fences.

## A.9 Text-to-SQL — Responder SELECT

SYSTEM

You translate a responder's natural-language query into a single safe SQLite SELECT against the request/offer tables. You may not modify data, may not query unregistered tables, and may not use literal values the user did not supply.

USER

```
SCHEMA: {DDL}
QUERY: {nl_query}
```

Forbidden: INSERT, UPDATE, DELETE, DROP, ALTER, ATTACH, PRAGMA, joins against unregistered tables.

Return JSON:

```
{
  "sql": str,          -- SELECT with ? placeholders
  "params": list[str],
  "explanation": str    -- one-sentence gloss
}
```

## B Schema Definitions and Data Structures

### B.1 Extraction Record (Pydantic)

```
class ExtractionRecord(BaseModel):
```

```
    request: bool
    offer: bool
    request_conf: float = Field(ge=0, le=1)
    offer_conf: float = Field(ge=0, le=1)
    resource_type: str = ""
    quantity: str = ""
    location: str = ""
    time: str = ""
    urgency: Literal["immediate", "high",
                    "moderate", "low", "unknown"]
    special_req: list[str] = []
    items_original: list[str] = []
    items_needs_clarification: list[str] = []
```

### B.2 Database DDL

CrisisLink persists into two structurally identical tables, request\_records and offer\_records, joined at matching time by taxonomy path and location.

```

CREATE TABLE request_records (
  id          INTEGER PRIMARY KEY,
  post_id     TEXT NOT NULL,
  post_text   TEXT,
  item        TEXT NOT NULL,
  quantity_raw TEXT,
  quantity_norm REAL,
  quantity_unit TEXT,
  location_raw TEXT,
  location_geo TEXT,
  time_raw    TEXT,
  time_iso    TEXT,
  urgency     TEXT,
  special_req TEXT, -- JSON array
  segment     TEXT,
  family      TEXT,
  class_name  TEXT,
  grounding_conf REAL,
  criticality REAL,
  clarif_state TEXT, -- pending / done / triage
  provenance  TEXT, -- JSON object
  source_file TEXT,
  UNIQUE (post_id, item)
);

CREATE INDEX r_idx_item    ON request_records(item);
CREATE INDEX r_idx_segment ON request_records(segment);
CREATE INDEX r_idx_urgency ON request_records(urgency);
-- offer_records: identical schema, separate table.

```

### B.3 Curated UNSPSC Subset

The curated taxonomy used at runtime contains 58 segments, 658 families, and 8,637 classes, drawn from the full UNSPSC catalog of 149,846 classes. Coverage was decided in a two-pass review: an LLM assigned an `is_relevant` flag to every node, after which the human reviewer confirmed or overrode the LLM judgment with a cascade rule (rejecting a family rejects all its classes; accepting a class lifts its family into the reviewed set).

## C Annotation Protocol

### C.1 Annotators and Workflow

Two trained annotators reviewed each component. Annotators were first calibrated on a 100-record set drawn from a held-out sub-corpus, on which they iterated written guidelines to convergence. For the reported audits, the two annotators independently reviewed the outputs produced by CrisisLink and judged whether each output satisfied the task-specific correctness criteria. Disagreements were surfaced through an audit UI, discussed between annotators, and resolved to consensus; a third reviewer was used only in deadlock ( $< 3\%$  of items in any stage). Cohen’s  $\kappa$  is computed on the two independent judgments prior to reconciliation, while the consensus labels are used only to compute the final accuracies.

### C.2 Per-stage Guidelines (excerpt)

**Offer/Request Classification.** Annotators were shown the request/offer labels predicted by CrisisLink and judged whether each label was correct. A post is *request* only if it expresses an explicit ask for a concrete, named resource. Rhetorical asks (“*we need a miracle*”) and pure situational reports (“*the road is blocked*”) are not request. Symmetric rules apply to *offer*.

#### Item Extraction.

Annotators were shown the item extracted by CrisisLink and judged whether the extracted item correctly reflected the resource expressed in the source post. When a generic term was extracted, annotators additionally judged whether the post supported a more specific item description.

#### UNSPSC Grounding.

The annotator reviews the UNSPSC path proposed by CrisisLink and judges whether the assigned segment, family, and class are valid and sufficiently specific given the source post. Where the LLM suggests a class and the annotator believes only family is warranted, the depth is downgraded. The depth-2 fallback is always considered valid; class-level labels are validated against the curated subset.

**NL→SQL.** A statement is correct iff (i) it parses under the structural whitelist, (ii) its bound parameters reflect the source post’s values, and (iii) it inserts/queries the registered table.

**Matching.** Annotators were shown the request record, the offer record proposed by CrisisLink, and the matching mode/rationale. A match is *correct* iff the offer can operationally satisfy the request given the recorded constraints. Sibling-class matches require that the substitution score is non-blocked and that the substitution rationale is acceptable to the annotator on review.

**Persona cards and role-players.** Clarification replies were produced by three volunteer lab members reading persona cards that specify the disaster context and the poster’s knowledge and behaviour, to induce realistic reply variability.

**Pair construction.** The 1,500 pairs comprise 1,000 corpus-derived and 500 controlled simulation pairs. Requests are sampled at random from the filtered donation-related subset; for each, CrisisLink returns its top-ranked offer from the real offer pool. Where no suitable real offer exists, we construct a controlled offer together with hard

distractors drawn from the same UNSPSC family. Each request is run under both conditions (inference only; one clarification round), and annotators judge the returned offer, not a pre-selected gold pair.

## D Hyperparameters and Policy Constants

| Constant                      | Value             |
|-------------------------------|-------------------|
| LLM backbone                  | DeepSeek-V3.2     |
| Decoding temperature          | 0.0               |
| Max parse retries             | 3                 |
| Pydantic-validated outputs    | all stages        |
| Clarification rounds per post | $\leq 3$          |
| Max questions per round       | 2                 |
| MCQ option count              | 3–9               |
| Response window               | 30 min (pilot)    |
| Per-poster back-off window    | 24 h              |
| Inference confidence floor    | 0.70              |
| Inference low-conf tag        | $< 0.70$          |
| UNSPSC seed frequency         | $> 30$            |
| PARENT_CONF_MIN               | 0.80              |
| CLASSIFY_CONF_MIN             | 0.30              |
| Service radius (default)      | 25 km             |
| Top- $k$ matches              | 5                 |
| Substitution score range      | $[0, 1]$          |
| Hard-block override           | rejects cover     |
| INSERT placeholder count      | 18                |
| SQL validator                 | sqlglot           |
| Duplicate policy              | INSERT OR REPLACE |

## E Category-to-Priority Mapping

The clarification policy ranks gap attributes by operational priority, conditional on the resource category. The mapping table below is supplied to the question-generation prompt (Appendix A).

| Category            | Priority order                   |
|---------------------|----------------------------------|
| Food / water        | quantity > location > urgency    |
| Shelter             | quantity > location > duration   |
| Medical: blood      | blood-type > recipient > qty     |
| Medical: medication | formulation > strength > qty     |
| Medical: general    | specific resource > location     |
| Rescue / SAR        | location > headcount > hazard    |
| Logistics           | route > vehicle type > qty       |
| Hygiene             | quantity > age/gender > location |

The mapping is consulted only to seed the question generator’s ordering; the generator may deviate when one attribute is already present at high confidence.

## F Worked Examples

### F.1 Plasma Case (extended, §2.3)

The walkthrough in §2.3 shows the surface flow. The full provenance trail recorded for that record contains:

```

post_id      : t_n15_00427
event       : 2015 Nepal earthquake (day 1)
post_text   : "Severe burns patient at Patan
              Hospital, need plasma immediately,
              please help."
```

```

extraction   : {request: true, resource: plasma,
               quantity: "", location:
               "Patan Hospital",
               urgency: "immediate"}
grounding    : depth=2, segment="Medical tissues
               and organs", family="Blood and
               blood products", class=NULL
gap_check    : present={resource_type, location,
               urgency}, missing={quantity},
               underspecified={ABO/Rh type,
               recipient capability}
clarification: Q1 "Blood type required?",
               Q2 "Medical staff on site?"
response     : Q1=0-, Q2=nurse
re-grounding : depth=3, class="Blood plasma"
re-check     : pass
persisted    : INSERT into request_records (...)
matched      : offer #o_blood_0913, 0-,
               cold-chain capable, exact mode
```

### F.2 No-reply Case

A post from the 2015 Nepal earthquake corpus, “*Mom is still trapped under the rubble, we need rescue equipment now please!!*”, reaches clarification with gaps {location, quantity} and underspecified resource. The clarification window closes without a reply. Inference recovers location=Sankhamul (provenance: an earlier post by the same poster in the thread, conf. 0.78) and urgency=immediate (provenance: in-text cues, conf. 0.91), but resource\_type remains *rescue equipment* (generic), and quantity remains missing. The record is routed to manual triage with both unresolved attributes flagged. Inference does not commit to *hydraulic cutters* on the strength of the rubble context.

## G Failure-mode Catalogue

We coded 200 end-to-end failures qualitatively and surface four recurring patterns, with illustrative examples:

**Implicit-resource posts.** “*The tap has been dry for two days.*” The extractor cannot recover **water** without the world knowledge that a dry tap implies a water need. Extraction confidence is low and the record is filtered.

**Rhetorical / sarcastic asks.** “*we need a miracle*”, “*send prayers*”. Pass REQUEST classification but yield no recoverable resource; downstream disambiguation marks the post for triage rather than synthesizing a resource.

**Out-of-taxonomy brand names.** A post asking for *NovoRapid* (a rapid-acting insulin brand) currently degrades to the *Drugs and pharmaceutical products / Insulin preparations* family because the curated subset lacks brand-level classes. The user-additions extension can fix this case but was not

exercised in the audited run.

**Multi-event threads.** When a thread spans two distinct events (e.g., an ongoing flood plus a separate medical emergency posted by the same account), inference can pull context from the wrong event. The current implementation has no thread-disambiguation step and treats nearby posts uniformly; this is the most common source of unrecoverable location errors.