TextMine: LLM-Powered Knowledge Extraction for Humanitarian Mine Action

Anonymous ACL submission

Abstract

Humanitarian Mine Action has generated extensive best-practice knowledge, but much remains locked in unstructured reports. We introduce **TextMine**, an ontology-guided pipeline that uses Large Language Models to extract knowledge triples from HMA texts. TextMine integrates document chunking, domain-aware prompting, triple extraction, and both reference-based and LLM-as-a-Judge evaluation. We also create the first HMA ontology and a curated dataset of real-world demining reports. Experiments show ontology-aligned prompts boost extraction accuracy by 44.2%, cut hallucinations by 22.5%, and improve format conformance by 20.9% over baselines. While validated on Cambodian reports, TextMine can adapt to global demining efforts or other domains, transforming unstructured data into structured knowledge.

1 Introduction

Landmine clearance remains a critical humanitarian challenge. In 2022 there were 4,710 casualties globally, 85% civilians (United Nations, 2025; Inclusion, 2023). Mine action authorities worldwide publish technical reports, surveys, lessons learned. However, this demining knowledge is largely underutilized due to unstructured formats and the absence of an integrated demining knowledge base. Extracting structured knowledge in the form of a knowledge graph from these reports allows sharing, accessing, and learning from the recorded experiences. It will provide actionable insights to improve decision-making and operational efficiency in humanitarian demining.

Cambodia has suffered over 65,000 casualties since 1979 (United Nations Development Programme (UNDP), 2021). In collaboration with the Cambodian Mine Action Centre (CMAC), we aim to convert their technical reports into a structured knowledge base. An example context and corresponding extracted triples under our HMA ontology is provided in Figure 1.

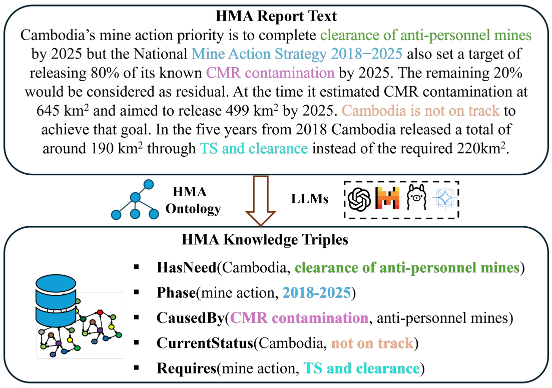

Figure 1: Example of extracting KG triples from humanitarian mine action (HMA) texts, guided by specific ontology. The task is to extract as many triples as possible while ensuring conformance to the ontology and faithfulness to the source text.

Constructing knowledge graphs from unstructured text remains resource–intensive. Traditional pipelines, such as Named Entity Recognition followed by Relation Extraction, require separate models and often domain-specific fine-tuning (Nadeau and Sekine, 2009; Pawar et al., 2017). Task-independent LLMs can automate both subtasks without additional training and match or exceed fine-tuned models (Wang et al., 2023; Jinensibieke et al., 2024). Recent benchmarks like Text2KGBench (Mihindukulasooriya et al., 2023) and ontology-guided sentence-level methods (Cauter and Yakovets, 2024) have demonstrated promise, but they rely on toy ontologies and single-sentence inputs, limiting their applicability to complex, domain-specific documents.

Unlike prior sentence level approaches, TextMine reasons over entire paragraphs, enabling coreference resolution and multi step inference, while relying on a practical operational HMA ontology that is orders of magnitude larger than those used in existing benchmarks (Table 1).

Table 1: Comparison of TextMine with other triple extraction methods. "Operational Ontology" denotes active use in real world applications.

Criteria	Text2KGBench	(Cauter and Yakovets, 2024)	TextMine
Domain	General	Maintenance	HMA
Input Granularity	Sentence	Phrase	Paragraph
Operational Ontology	✗	✓	✓
Paragraph Level	✗	✗	✓

One key challenge in our work is to investigate effective methods to activate LLMs to reason over paragraph-level text while adhering to a given ontology. Another major challenge is to establish a reliable evaluation framework for the extracted triples. We summarize the contributions of this work as follows:

- **Automated Demining Knowledge Extraction Framework.** We introduce TextMine, a prompt-based pipeline that combines layout-aware document chunking, ontology-guided extraction, and multi-perspective evaluation. To our knowledge, this is the first LLM application of knowledge extraction in the HMA domain.
- **HMA Ontology and Evaluation Dataset.** We introduce a dedicated ontology for humanitarian demining operations, systematically categorizing operational entities and relationships. Alongside this ontology, we create a human-annotated, ontology-aligned triple dataset, filling a critical resource gap for this domain.
- **Reference-Based and Reference-Free Evaluation.** We evaluate extracted triples against our annotated dataset and introduce a bias-aware LLM-as-Judge framework for reference-free scoring. Experiments on closed and open LLMs show that position bias skews rankings, which randomizing output order mitigates for some models.
- **In-Context Learning Optimization.** We examine demonstration design and find that prompts enriched with ontology-aligned examples improve triple extraction accuracy by up to 44.2%, reduce hallucinations by 22.5%, and enhance format conformance by up to 20.9% compared to baseline prompts. These findings provide practical insights for prompt construction.

2 The Humanitarian Mine Action

Demining operations in Cambodia incur costs in the billions (Harris, 2000), yet most agencies still manage data via spreadsheets and manual review, which impedes timely analysis and collaboration. In partnership with CMAC, we address this by automating knowledge extraction from both field and technical reports (Figure 2). Our framework serves as a proof of concept for LLM driven demining knowledge extraction, transforming unstructured reports into structured insights that can directly inform and optimize future clearance planning, and is readily adaptable to other mine affected regions.

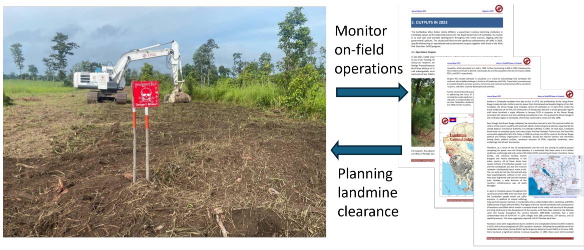

Figure 2: Left: CMAC landmine clearance operation. Right: CMAC Technical Progress reports. Our workflow extracts knowledge from demining reports with the aim to leverages it to guide future clearance planning.

2.1 Task Description

Knowledge triple extraction from HMA documents can be formally defined as follows: Given an ontology $\mathcal{O} = (\mathcal{E}, \mathcal{R})$, where $\mathcal{E}$ represents ontology entities and $\mathcal{R}$ represents ontology relations, and a textual context C, the objective is to design a prompt $P(C, \mathcal{O})$ that guides an extractor model M to extract triples $T = \{(s_1, r_1, o_1), (s_2, r_2, o_2), \cdots, (s_n, r_n, o_n) \mid s, o \in \mathcal{E},\ r \in \mathcal{R}\}$, where n is the number of extracted triples ($n = |T|$), subjects and objects from source text $s, o \in \mathcal{E}$ (mapped to the entities $\mathcal{E}$ in the ontology) and relations from the ontology $r \in \mathcal{R}$. The extracted triples must remain consistent with both the ontology and the source text. Shortly, this problem can be written as $T = M\big(P(C, \mathcal{O})\big)$.

3 TextMine Overview

Figure 3 shows our triple extraction method. In the **Layout-Aware Document Chunking** phase, PDF reports are split into paragraph chunks. These are used with a newly constructed HMA ontology in the **Ontology-Guided Knowledge Extraction** phase. For evaluation, we apply a **Multi-Perspective Evaluation** combining reference-based metrics on our annotated dataset and a reference-free LLM-as-a-Judge approach. All LLM calls use greedy decoding (temperature = 0, top_p = 1.0) to ensure deterministic outputs.

3.1 Layout-Aware Document Chunking

The input to our pipeline is PDF-formatted demining reports, which contain rich human-readable

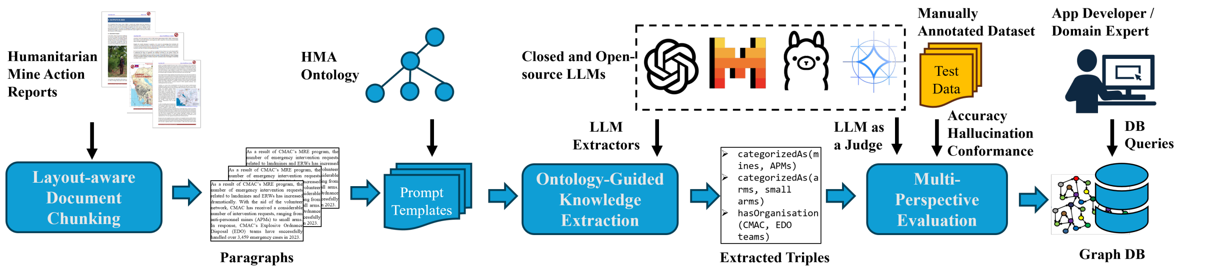

Figure 3: **TextMine Overview.** Reports are preprocessed into paragraph chunks, used as test inputs during inference. Each chunk is combined with an instruction template and ontology, then passed to LLMs for triple extraction. We apply a multi-perspective evaluation using both reference-based and reference-free methods. Extracted triples can be stored in a Graph DB for querying by developers and domain experts.

structures (chapters, sections, tables, lists, and figures). To prepare them for LLM consumption, we segment each document into semantically coherent chunks that preserve context while fitting within typical model context windows (Liu et al., 2024). We leverage Open-Parse's document understanding capabilities (Smock and Pesala, 2021) to identify layout elements and extract paragraph-level segments. On our reports, this yields chunks averaging 127 words (std. 6), which aligns well with both small and large LLM context limits.

3.2 Ontology-Guided Knowledge Extraction

Given the text chunks as input, our goal is to extract triples $T = \{(s_1, r_1, o_1), \cdots, (s_n, r_n, o_n) \mid s, o \in \mathcal{E},\, r \in \mathcal{R}\}$, where $s, o \in \mathcal{E}$ (subject and object pairs) and $r \in \mathcal{R}$ (relations). First, the extracted triples must be accurate w.r.t., the source text. An ideal extraction system would extract all triples (recall=1) precisely (precision=1). Second, extracted triples must conform to a specified ontology so that extracted demining knowledge can be stored and shared between organizations and countries in a compatible way. We address this by combining a domain-specific HMA ontology with LLM in-context learning to extract triples from paragraphs.

HMA Ontology We performed a survey of demining related ontologies together with domain experts and we incorporate six ontologies from IMSMA[1]. These ontologies are used for demining information system but not strictly for HMA, We then manually filter out concepts and relations unrelated to HMA. Furthermore, we add a more general humanitarian domain ontology from *Empathi* (Gaur et al., 2019) to make the overall HMA ontology more comprehensive. As a result, HMA ontology integrates seven ontologies (160 entity, 86 relation types) covering diverse aspects of humanitarian demining action (see Table 2).

Table 2: Number of entity types and relation types for each ontology.

	Empathi	Assistance	Accident	Activity	Land	Location	Victim
Entity Types	20	14	34	29	31	5	27
Relation Types	22	7	6	24	10	5	12

Prompt Templates We design five prompting strategies for knowledge triple extraction: (1) *Zero-shot* (instruction and context only), (2) *One-shot with Random Sentence (RS)*, (3) *One-shot with Random Paragraph (RP)*, (4) *One-shot with Ontology-Aligned Sentence (OS)*, and (5) *One-shot with Ontology-Aligned Paragraph (OP)*. One example prompt is provided in Appendix 6.

Ontology-aligned demonstrations share the same ontology (entity types and relationships) as the target context, while *random* demonstrations use unrelated ontology. Sentences are extracted from paragraphs using NLTK's sentence tokenizer[2]. For OS-/OP prompts, we design a retrieval algorithm that selects demonstrations from our annotated dataset (Section 4.1) by identifying the shortest context-answer pair matching the target ontology. This is for minimizing token costs while retaining high semantic similarity. To prevent data leakage and ensure a fair evaluation, we implement a second retrieval step if the initially retrieved demonstration contains the same or part of target context as the test instance. In such cases, we select the next shortest matching example instead. This ensures that retrieved demonstrations do not overlap with the evaluation context, preserving the integrity of the inference.

These prompt templates are designed to test our two **hypotheses**: (a) *Ontology alignment enhances*

[1]Information Management System for Mine Action (IMSMA), https://www.gichd.org/our-response/information-management/imsma-core/

[2]https://www.nltk.org/howto/stem.html

accuracy by priming the model with ontology-specific reasoning. Semantically aligned demonstrations help constrain the label space and improve precision (Min et al., 2022; Long et al.). This effect is evident in our results as the contrast between RS/RP and OS/OP confirms the benefit of ontology alignment. (b) *Paragraph-level context improves extraction performance* by providing richer demonstrations that reflect how entities and relations are introduced across sentences in real-world reports. However, our results do not support this hypothesis as comparisons between RS vs. RP and OS vs. OP show no consistent improvement from paragraph-level context.

3.3 Multi-Perspective Evaluation

Reference-based Evaluation Humanitarian Demining is a high-stakes decision-making domain, incorrect triples, especially hallucinated ones, can misinform demining operations, leading to safety risks. A comprehensive evaluation of triple extraction requires assessing accuracy, reliability, and structural validity. We evaluate models across three dimensions: (1) Triple Extraction Accuracy, (2) Hallucination Rate, and (3) Format Conformance.

Triple Extraction Accuracy We employ N-gram matching-based metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005) to assess the extracted triples. Additionally, we incorporate BERTScore (Zhang et al., 2019), which leverages word embeddings to capture semantic similarity beyond lexical overlap. To enhance the accuracy of our metrics, we first apply stemming and lemmatization using NLTK to normalize morphological variations. We then compute the accuracy metrics against the manually annotated test set.

Triple extraction accuracy serves as the primary metric, as it directly measures how well models extract knowledge triples from text. However, accuracy alone does not fully capture model reliability. The hallucination rate evaluates faithfulness by detecting extraneous or fabricated information, while format conformance ensures that outputs adhere to a syntactically valid structure, enabling seamless integration into downstream applications.

Hallucination Rate Accuracy primarily measures how well the extracted triples match reference triples, but it does not fully assess whether the generated content is grounded in the input. A model could produce plausible triples that are semantically similar to the original text (thus achieving a high accuracy score) yet still be incorrect, meaning they do not appear in the input but seem reasonable. This distinction is crucial in high-stakes applications like humanitarian demining, where misleading information can have serious consequences. Hallucination is a prevalent issue in LLM-generated content (Ji et al., 2023; Huang et al., 2023; Xu et al., 2024) and a critical aspect of our evaluation. To quantify this, for each extracted triple $t = (s, r, o)$ we first normalize s, r, and o, as well as the entire input context (tokenization, lemmatization, lowercasing, and punctuation removal). We then check whether the normalized subject s and object o occur as contiguous substrings in the normalized report text—if not, t is flagged as a hallucination. Similarly, if the normalized relation r is not found in the normalized ontology relation set, t is also flagged. This procedure ensures that even "plausible" but unsupported triples are detected and penalized, thereby maintaining faithfulness and trustworthiness in our system.

Format Conformance Format conformance metric assesses whether the generated triples adhere to the correct syntactic format of $r(s, o)$, where r represents the relation and s and o denote the subject and object, respectively. We consider a triple well-formatted if it follows this structure. We accommodate edge cases where the subject or object contains numerical values with commas, such as `hasReliabilityInfo(2,500,011 square meters, landmine/ERW affected areas)`, or phrases in parentheses, such as `hasAccidentOrganisationInfo(Quality of Life Survey (QLS), Department of Victim Assistance of CMAA)`. Format conformance ensures that extracted triples follow a structured format necessary for practical use. A model with high accuracy but poor format conformance may fail to produce usable outputs, limiting its applicability in real-world knowledge extraction tasks.

Combined Score To unify evaluation metrics into a single representative score, we apply min-max normalization and compute the overall *Combined Score* as:

$$S_{\text{combined}} = \frac{1}{k}\big(S'_{\text{BLEU}} + S'_{\text{ROUGE}} + S'_{\text{METEOR}} + S'_{\text{BERTScore}} + (1 - S'_{\text{Hallucination}})\big) \quad (1)$$

where S' represents the normalized metric values within $[0,1]$, and the hallucination rate is inverted to penalize higher hallucination. $k=5$ is the number of metrics included in the score. Format conformance is excluded in the Combined Score, as our experimental results show consistently high format conformance across all extraction models and prompt settings, making it non-differentiating. *Combined Score* provides a holistic measure of extraction quality while mitigating scale differences among individual metrics.

Reference-Free Evaluation Evaluating generated texts is particularly challenging in domains like humanitarian demining, where annotated datasets are scarce. Demining reports are highly technical and domain-specific, requiring extracted triples to align with predefined ontologies of landmine types, clearance operations, and affected areas. Constructing a manually labeled test set is time-consuming and resource-intensive, limiting large-scale reference-based evaluation. Given these constraints, we explore an LLM-as-a-Judge approach as a potential reference-free evaluation framework for evaluating extracted triples.

LLM-as-a-Judge offers a potential alternative to evaluation when ground-truth data is limited (Friel and Sanyal, 2023; Saad-Falcon et al., 2023; Es et al., 2023). The ultimate objective of our approach is to find a optimal judge LLM setting where the LLM consistently identifies the best candidate answer and provides a reasoned justification for its decision.

We try to find the optimal LLM Judge setting by conducting systematic ranking experiments and analyzing correlations between the LLMs judged ranking and reference-based rankings. For these ranking experiments, we design Judge Prompts that instruct LLMs on evaluation criteria. We use five models including Mistral-7B[3], Llama3-8B[4], Gemma2-9B[5], LLaMA3-70B[6] and GPT-4o[7] as extraction models. We rank five responses from five models using GPT-4o, Llama3.1-70B[8], and Llama3.3-70B[9] as our judge models.[10] The judge prompts follow a fixed template with seven placeholders, where the ontology placeholder represents entity and relation types. Formally, let the input set for the LLM judge prompts be $\{O, C, R_{m_1}, R_{m_2}, R_{m_3}, R_{m_4}, R_{m_5}\}$, where O is the ontology set, C is the set of test contexts, and $R_{m_1}, \cdots, R_{m_5}$ are the five sets of candidate answers from the five extractor models. The judge LLM produces a verdict (output as a ranking):

$$V = \mathrm{LLM}\big(\{O, C, R_{m_1}, R_{m_2}, R_{m_3}, R_{m_4}, R_{m_5}\}\big), \tag{2}$$

assigning a rank from best (1) to worst (5) based on predefined instructions and ranking criteria.

To mitigate evaluation biases, we design three judge prompt templates: (1) Basic Judge Prompt, (2) Fair Judge Prompt, and (3) Randomized Fair Judge Prompt. These templates differ in their instructions and ranking methodologies. Fair Judge Prompt enforces explicit reasoning criteria to mitigate position bias, a known issue when LLMs evaluate multiple candidate answers simultaneously (Li et al., 2024; Shi et al., 2024). Randomized Fair Judge Prompt further reduces this bias by randomizing the position of candidate answers, ensuring that response order does not influence rankings.

Once the optimal judge LLM setting is determined, we adopt it as the reference-free evaluator to identify the best answer from each extraction, leveraging its reasoning process. Detailed prompt templates and an example of the reasoning process used for evaluation are provided in Appendix 6.

4 Experimental Results

We assess the effectiveness of our knowledge triple extraction method through *reference-based* and *reference-free evaluations*. Reference-based evaluation compares extracted triples against our curated dataset, while reference-free evaluation relies on LLM judges to assess generated triples without reference data. A key aspect of our analysis is examining the correlation between these two evaluation paradigms to evaluate the reliability of LLM-based judgments. Additionally, we investigate how different prompt strategies influence extraction performance across models.

4.1 Test Dataset

We constructed a dataset based on six recent mine action reports from CMAC: the 2023 Annual Report, the 2023–2024 Integrated Work Plans, the

[3] https://ollama.com/library/mistral:7b
[4] https://ollama.com/library/llama3:8b
[5] https://ollama.com/library/gemma2:9b
[6] https://ollama.com/library/llama3:70b
[7] https://openai.com/index/hello-gpt-4o/
[8] https://ollama.com/library/llama3.1:70b
[9] https://ollama.com/library/llama3.3:70b
[10] In our experiments, GPT-4o is used via the Azure OpenAI API, all the other open-source models are used via local deployment using Ollama.

2023 Cluster Munition Remnants Clearing Report, the 2023 Mine Clearing Report, and the Article 7 Report. In total, we processed 270 pages, yielding 549 text chunks. These text chunks are used as test instances in zero-shot prompts. We then generate model responses using GPT-4o and Llama3-70B for these zero-shot prompts. 100 prompt-response pairs are randomly sampled from them. The sampling ensures a balanced distribution across ontology categories.

The annotation workflow involved a human annotator reviewing the model-generated triples and discarding those that were semantically or factually incorrect. To prevent any bias, the annotator was kept blind to which model produced each output, ensuring that validation decisions did not favor a particular system. The remaining valid triples were then aggregated and de-duplicated to ensure uniqueness. This process resulted in a final dataset of 1,095 unique triples derived from 100 sampled prompt-response pairs. This curated dataset serves as a reliable test set for model evaluation by minimizing noise and redundancy while maintaining ontology diversity.

In the retrieval-based one-shot prompt template construction phase, some examples were retrieved from this dataset to construct demonstrations. These retrieved examples were excluded from being sampled into the final dataset to ensure an unbiased evaluation.

4.2 Reference-Based Evaluation

We employ five LLMs as extractor models: Llama3-70B, GPT-4o, Gemma-9B, Llama-8B and Mistral-7B. Figure 4 illustrates the impact of model selection and prompt strategy on extraction performance. The box plot (top) shows the distribution of *Combined Scores* across models. Llama3-70B achieves the highest overall performance score, closely followed by GPT-4o. Gemma2-9B demonstrates moderate performance, while Llama3-8B and Mistral-7B receive lowest overall scores. The line plot (bottom) highlights prompt effects, with OS and OP yielding the highest scores across most models, supporting the effectiveness of ontology-aligned prompting. These findings reinforce our hypothesis that ontology-aligned prompts enhance extraction accuracy.

Figure 5 further breaks down the performance of five models across four accuracy evaluation metrics: BLEU, ROUGE, METEOR, and BERTScore. OS demonstration prompts consistently result in

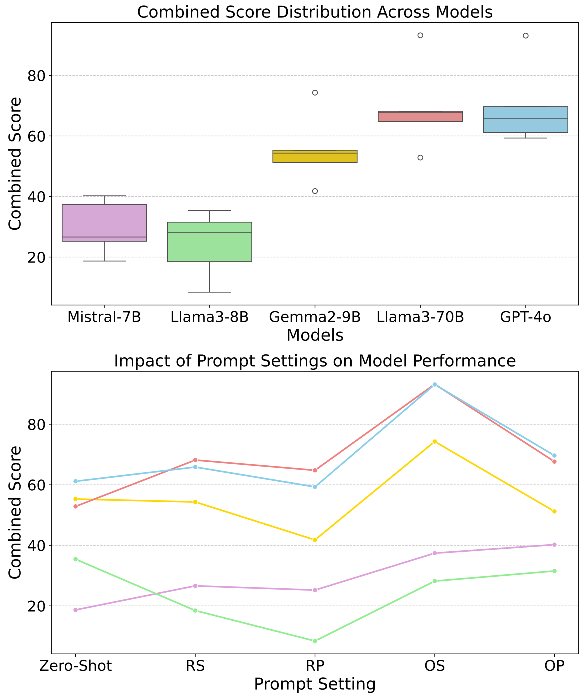

Figure 4: The combined visualization illustrates the impact of model selection and prompt strategy on extraction performance. The top Combined Score is achieved by Llama3-70B (93.24) closely followed by GPT-4o (93.13), both with OS prompt setting.

the best accuracy across all four metrics and all five models, highlighting their effectiveness for the triple extraction task. BLEU scores peak with OS prompts, with GPT-4o achieving the best performance. ROUGE results show a similar trend, with GPT-4o and Llama3-70B excelling in the OS prompt setting. METEOR follows the same pattern as BLEU and ROUGE, reinforcing the advantages of OS prompts. BERTScore, which measures semantic similarity, shows high clustering across models, suggesting minimal differentiation in performance. Overall, OS demonstration prompts consistently enhance extraction accuracy across all models and metrics.

Figure 6 presents the hallucination rates for subjects, relations, and objects across different models and prompt settings. GPT-4o, Llama3-70B, and Gemma2-9B exhibit lower hallucination rates for subjects and objects, while Llama3-8B and Mistral-7B tend to have higher hallucination rates. For high-performing models, the OS prompt type generally helps reduce hallucination, whereas RP prompts tend to increase it. Zero-Shot prompts often lead to increased hallucination for subjects and objects across models. Interestingly, however, Zero-Shot prompts show lower hallucination rates for rela-

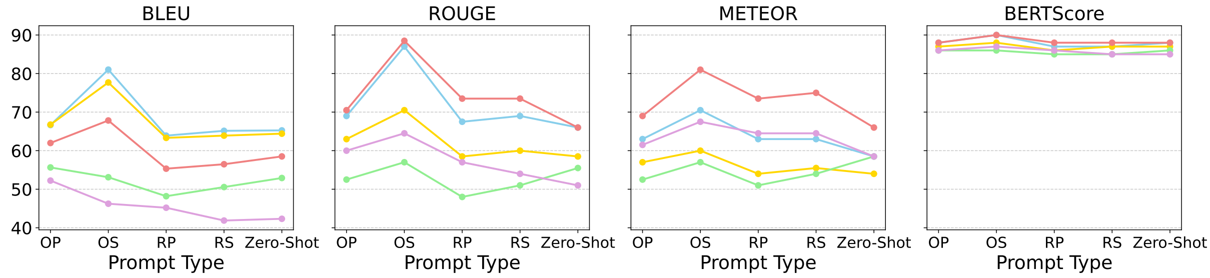

Figure 5: Accuracy metrics scores across prompt types for each model. OS demonstration prompts consistently result in the best accuracy across all four metrics and all five models. *Note: ROUGE, METEOR are scaled by 150, BERTScore by 100 for better visibility.*

tions, which may be due to the additional demonstrations in other prompts introducing noise that negatively impacts relation extraction.

Most models exhibit high format conformance across all prompt types, with only Gemma-9B under the Zero-Shot prompt scoring below 80%. GPT-4o and Llama3-70B consistently achieve FC above 95% across all prompt types, demonstrating superior adherence to the expected format.

4.3 Reference-Free Evaluation

Reference-based evaluation shows that the OS prompt consistently achieves the highest performance across most models, so for the reference-free ranking experiments we only focus on ranking the five models under OS prompt setting. To assess the alignment between reference-based and reference-free rankings, we compute correlations between the rankings derived from the *Combined Score* calculated based on references, and the rankings derived from the *Expectation Score* based on LLM judges.

Expectation Score As each extractor model received multiple rankings from different judge models, we compute a single *Expectation Score* per extractor model m. This score is defined as:

$$\textit{Expectation Score } E(m) = \frac{\sum_{i=1}^{\mu} \left(i \times P_m(i)\right)}{\sum_{i=1}^{\mu} P_m(i)} \quad (3)$$

where i represents a specific rank, $P_m(i)$ denotes the number of times the model m was assigned rank i. In our case from $i = 1, \cdots, \mu$ and $\mu = 5$ for the five extractor models. E provides a weighted average that reflects the overall tendency of judge models to place an extractor model at a particular rank. The extractor models are ranked such that the one with lowest *Expectation Score* is ranked the highest and vice versa. To systematically analyze the consistency and reliability of the judge methods, we compute the *Expectation Scores* of extractor models separately for Basic Judge, Fair Judge, and Randomized Fair Judge.

Correlation Between LLM Judged and Reference-Based Rankings To assess the reliability of LLM judges, we compute Spearman's correlation (ρ) and Kendall's Tau (τ) to quantify the alignment between LLMs judged rankings and reference-based rankings. ρ measures the monotonic relationship between rankings, where values close to 1 indicate strong agreement. τ evaluates ranking concordance by analyzing the number of concordant and discordant rank pairs, making it particularly useful for detecting minor positional changes. The results of our iterative ranking experiments, shown in Table 3, demonstrate how different judge methods impact ranking alignment.

Our findings reveal that introducing randomization significantly enhances ranking consistency for GPT-4o, improving from $\rho = 0.4$ (Basic) to $\rho = 1.0$ (Randomized). In contrast, Llama3.1-70B shows no improvement across judge methods, indicating persistent positional bias. Llama3.3-70B exhibits weaker alignment overall, with minor improvements under Fair and Randomized judging. These results suggest that while randomization effectively mitigates positional bias for GPT-4o, its impact varies across models.

Table 3: Correlation values for different judge models and judge methods.

Judge Model	Judge Method	Spearman's Correlation	Kendall's Tau
GPT-4o	Basic	0.4	0.4
	Fair	0.9	0.8
	Randomized	1.0	1.0
Llama3.1-70B	Basic	0.4	0.4
	Fair	0.4	0.4
	Randomized	0.4	0.4
Llama3.3-70B	Basic	0.0	-0.2
	Fair	0.3	0.2
	Randomized	0.2	0.2

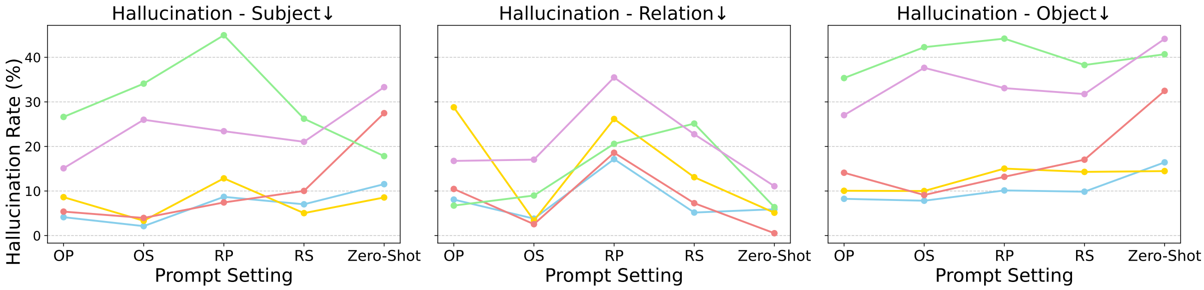

Figure 6: Hallucination rate of subject, relation, and object across prompt types for each model.

The ranking experiments identify GPT-4o with Randomized Fair Judge as the optimal judge LLM setting for our evaluation task. We further apply this setting to identify the optimal triples from each extraction output using GPT-4o with the Randomized Fair Judge method, aggregating the top-ranked triples across all extractions to generate the final output. We compare these aggregated results against the test dataset, yielding a *Combined Score* of 83.93. This achieves 90% of the current best score 93.24 (see figure 4), reinforcing the effectiveness of our reference-free LLM-as-a-Judge paradigm as a viable alternative to conventional reference-based evaluation methods. Future research could refine this approach by exploring additional LLMs and judge strategies to approximate a even better judge LLM setting for knowledge triple extraction tasks.

5 Related Work

The emergence of LLMs such as those from the GPT family (e.g., GPT-4o[11]), Llama (Touvron et al., 2023), BLOOM (Workshop et al., 2022), and PaLM (Chowdhery et al., 2023), has significantly transformed the field of knowledge extraction from text. These models possess advanced language understanding and reasoning capabilities, making them well-suited for extracting knowledge from unstructured text, especially when paired with prompting techniques like in-context learning (ICL) (Brown et al., 2020).

ICL enables LLMs to learn new tasks by providing input-output demonstrations during inference. Depending on the number of examples provided, this can range from zero-shot (no demonstrations) to one-shot or few-shot learning (multiple demonstrations) (Min et al., 2022; Liu et al., 2023). This method enhances the models' ability to generalize from minimal data. Zhu et al. (2023) demonstrated the effectiveness of ICL through a virtual knowledge extraction task.

Mihindukulasooriya et al. (2023) introduced an approach that utilizes ontology guidance to extract knowledge from text. Their work highlights the potential of LLMs in extracting domain-specific knowledge constrained by ontological rules. In our study, we adapt this approach to the domain of humanitarian demining, employing a set of specialized ontologies to guide the extraction process.

Evaluating generated texts is a challenging task, especially when limited ground truth data are available. To address this problem, recent approaches include generating synthetic data to train an evaluator model (Saad-Falcon et al., 2023), annotating datasets using a human-in-the-loop methodology (Dagdelen et al., 2024), or leveraging strong LLMs as judges (Zheng et al., 2024; Bavaresco et al., 2024). Our work involves annotating extracted triples to create a evaluation dataset, applying LLMs as judges and analyzing the alignment between these two evaluation methods.

6 Conclusion

TextMine addresses the need for automated knowledge extraction from HMA reports by LLMs and domain ontologies to transform unstructured reports into structured knowledge triples.Our findings highlight the effectiveness of ontology-aligned prompts in improving extraction accuracy.The introduction of the HMA ontology and a human-annotated evaluation dataset bridges the gap in standardized resources for the HMA domain. Furthermore, our bias-aware reference-free evaluation using LLM-as-a-Judge demonstrates great potential.

[11] OpenAI GPT-4o, https://openai.com/index/hello-gpt-4o/

Limitations

While TextMine demonstrates the feasibility of an LLM-driven extraction pipeline in the specialized domain of humanitarian demining, several limitations warrant acknowledgement. First, our evaluation relies on just 100 prompt–response examples (yielding 1,095 unique triples); although these examples were selected to cover diverse geographies and operational contexts, the small scale constrains broad generalizability. Assembling and annotating demining data demands extensive domain expertise, so even this modest set offers valuable proof-of-concept insights, but we plan to extend to larger, multilingual collections in future work. Second, ground-truth triples were produced via a model-assisted, expert-validated workflow by a single annotator; the absence of multiple annotators prevents calculation of inter-annotator agreement, and future studies will involve multi-annotator labeling to quantify label reliability. Third, although the annotator was blinded to model provenance when validating candidate triples, using model suggestions as a starting point may introduce subtle bias toward the evaluated model family; to mitigate this, we intend to curate a purely human-authored gold set for a representative subset. Finally, while we acknowledge that in-depth analysis of the LLM-judge's own hallucination and bias characteristics is important, conducting a comprehensive bias and hallucination audit falls beyond this paper's core scope and is deferred to future work focused explicitly on judge calibration and fairness.

Ethical Considerations

Humanitarian demining is a high importance decision making domain where incorrect or hallucinated triples can misinform planning and lead to wasted time and resources. To address this, TextMine combines reference based validation with a bias aware LLM as Judge framework and publishes all prompts and decoding settings for full transparency. We also engage CMAC and ICRC landmine clearance experts throughout development to review and validate outputs. TextMine is not used to locate mines, as safety remains governed by established GICHD standards[12], and instead supports expert analysis and planning. By documenting our methods and keeping an expert in the loop, we aim to minimize misinformation and ensure responsible AI deployment in demining operations.

The dataset used in this study consists of publicly available or institutionally provided humanitarian demining reports. These reports were reviewed to ensure they do not contain personally identifiable information (PII) or offensive content. Where necessary, documents were anonymized or filtered to remove sensitive information. Our usage of the data adheres to privacy standards and is strictly confined to research contexts.

All datasets used in this study were accessed under conditions permitting research use. The curated demining report dataset we constructed is intended solely for academic and research purposes and complies with the original access and licensing conditions. The ontology and pipeline components developed in TextMine are likewise designed for research and evaluation within humanitarian domains. We do not support or promote deployment of these artifacts in operational or commercial contexts without further validation and ethical review.

[12]GICHD Mine Action Standards, https://www.gichd.org/our-response/mine-action-standards/

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, and 1 others. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *arXiv preprint arXiv:2406.18403*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zeno Cauter and Nikolay Yakovets. 2024. Ontology-guided knowledge graph construction from maintenance short texts. In *Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024)*, pages 75–84.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Robert Friel and Atindriyo Sanyal. 2023. Chainpoll: A high efficacy method for llm hallucination detection. *arXiv preprint arXiv:2310.18344*.

Manas Gaur, Saeedeh Shekarpour, Amelie Gyrard, and Amit Sheth. 2019. Empathi: An ontology for emergency managing and planning about hazard crisis. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 396–403.

Geoff Harris. 2000. The economics of landmine clearance: case study of cambodia. *Journal of international development*, 12(2):219–225.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Humanity & Inclusion. 2023. Landmine monitor 2023: Current conflicts & long-lasting contamination cause high number of mine casualties. Accessed: 29-Jan-2025.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Dawulie Jinensibieke, Mieradilijiang Maimaiti, Wentao Xiao, Yuanhang Zheng, and Xiangbo Wang. 2024. How good are llms at relation extraction under low-resource scenario? comprehensive evaluation. *arXiv preprint arXiv:2406.11162*.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024. Split and merge: Aligning position biases in llm-based evaluators. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11084–11108.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Quanyu Long, Yin Wu, Wenya Wang, and Sinno Jialin Pan. Does in-context learning really learn? rethinking how large language models respond and solve tasks via in-context learning. In *First Conference on Language Modeling*.

Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F Enguix, and Kusum Lata. 2023. Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text. In *International Semantic Web Conference*, pages 247–265. Springer.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

David Nadeau and Satoshi Sekine. 2009. A survey of named entity recognition and classification. In *Named Entities: Recognition, classification and use*, pages 3–28. John Benjamins publishing company.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. 2017. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476*.

Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms. *arXiv preprint arXiv:2406.07791*.

Brandon Smock and Rohith Pesala. 2021. Table Transformer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

United Nations. 2025. International mine awareness day.

United Nations Development Programme (UNDP). 2021. Clearing for Results Phase IV: Mine Action for Human Development. Accessed: February 9, 2025.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, and 1 others. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*.

A In-Context Learning Prompt Example

The below is an example of one-shot prompt with RS prompt setting. All in-context learning prompts are stored in CSV files as part of the supplementary material for easier reproducibility.

One-shot with Random Sentence Demonstration (RS)

Instruction:
Extract and list only the triples from the following sentence based on the specified entity types and relation types. Do not include any explanatory or intermediate text in your output. In the output, only include the triples in the given output format: relation(subject, object). Attempt to extract as many entities and relations as you can.
Entity Types:
AdministrativeArea, Association, Location, Organisation, MedicalFacility
Relation Types:
hasAdministrativeArea, hasAssociation, hasLocation, hasOrganisation, locatedNear
Example:
Sentence:
The accidental detonation of old wartime munitions causes significant infrastructure damage to the nearby village roads and buildings.
Output:
CausedBy(infrastructure damage, old wartime munitions)
Context: On Thursday, March 16, 2023, at CMAC Headquarters in Phnom Penh, Delegate of the Royal Government in charge as Director General of CMAC, met with a delegation from the Japan International Cooperation Agency (JICA) General Director of Governance and Peacebuilding Department. During the meeting, the JICA side briefed on the results of its cooperation with CMAC, in particular training for Ukraine with good results.

B LLM Judge Prompts

The below are two example prompts used during the experimental study: 1) Basic judge prompt, and 2) (Randomized) Fair Judge Prompt. These methods are explained in Sec. 3. The latter prompt example below includes both cases of regular and randomized fair judge prompts at once. The only difference is the shuffling of the positions of candidate answers in "Model Outputs" part of the prompt. All LLM judge prompts are stored in CSV files as part of the supplementary material. The prompts can be used for reproducing as well as applying in different datasets (without additional annotation efforts).

Basic Judge Prompt

Instruction:
You are a judge who ranks five models from 1 to 5 on a triple extraction task. You must assign 1 to the model with the best answer and 5 to the model with the worst answer. Your ranking should be provided directly in this format: [1: model x; 2: model x; 3: model x; 4: model x; 5: model x].
Ranking Criteria:
Correctness:
The triples must conform to the format relation(subject, object) and must accurately reflect relationships stated in the context. Models with significant formatting errors should be penalized.
Coverage:
The number of correct triples extracted. More accurate triples are better, but avoid penalizing slight redundancies unless they detract from the overall relevance.
Relevance:
The triples must be relevant to the specified entity and relation types and should align well with the specific context provided.
Edge Cases:
If a model extracts many triples but includes incorrect or redundant ones, balance accuracy and redundancy in your ranking. Correctness should be prioritized, followed by Relevance, then Coverage.
Entity Types: {entity_types}
Relation Types: {relation_types}
Context: {Context}
Model Outputs: {model 1 output} {model 2 output} {model 3 output} {model 4 output} {model 5 output}
Your ranking:

(Randomized) Fair Judge Prompt

Instruction:
You are a judge tasked with evaluating and ranking five models based on their performance in a **triple extraction task**. Your role is to ensure **fairness, impartiality, and accuracy** by independently evaluating each model's output without any positional bias. Do not assume that the first model is better or worse simply because of its position—all models must be treated equally.
Evaluation Guidelines:
1. **Independence of Evaluation**:
Evaluate each model **independently** without comparing it to others until all models are scored. Avoid assumptions based on position or order in the list.
2. **Evaluation Criteria:**
(a) **Correctness of Triples (Highest Priority):**
- Triples must strictly conform to the format `relation(subject, object)`.
- Relationships must match the **Given Relation Types** provided below.
- Triples containing fabricated or hallucinated relationships must result in a significant penalty.

(b) **Relevance:**
- Triples must accurately reflect relationships mentioned in the **Context**.
- Irrelevant triples or hallucinations must receive a lower score.

(c) **Coverage:**
- The number of correct triples extracted. Higher coverage is better **only** if triples meet correctness and relevance criteria.

3. **Ranking Process:**
- **Step 1:** Independently evaluate each model's output and assign scores (from 1 to 10) for each criterion: Correctness, Relevance, and Coverage. Summarize the total score for each model.
- **Step 2:** Rank all five models from 1 (best) to 5 (worst) based solely on their total scores.
- Break ties by prioritizing **Correctness** first, then **Relevance**, and finally **Coverage**.

Important Note:
Treat all models equally regardless of position. Your goal is to be an impartial judge. Avoid positional bias and ensure the evaluation reflects only the quality of the model outputs.
Output Format:
Provide your output strictly in this format:
Evaluation:
Model 1: Correctness = X, Relevance = X, Coverage = X, Total = Y
Model 2: Correctness = X, Relevance = X, Coverage = X, Total = Y
Model 3: Correctness = X, Relevance = X, Coverage = X, Total = Y
Model 4: Correctness = X, Relevance = X, Coverage = X, Total = Y
Model 5: Correctness = X, Relevance = X, Coverage = X, Total = Y
Ranking: [1: Model X; 2: Model X; 3: Model X; 4: Model X; 5: Model X]
Entity Types: {entity_types}
Relation Types: {relation_types}
Context: {Context}
Model Outputs: {model 1 output} {model 2 output} {model 3 output} {model 4 output} {model 5 output}
(For Randomized Fair Judge Prompts, the positions of candidate answers in Model Outputs are randomly shuffled.)
Your evaluation and ranking:

868

C Evaluation Reasoning Process

Output of GPT-4o with Randomized Judge Prompt for Extraction Prompt 89

Model's Answer for Prompt 89

Evaluation

Model 1

- **Correctness** = 4
  - Many relations are either incorrect or irrelevant to the context (e.g., Monitoring, Advises, Coordinates), leading to significant penalties.
- **Relevance** = 5
  - The model produces correct triples but includes several irrelevant triples not related to the context (e.g., Physical-Health).
- **Coverage** = 7
  - High number of triples extracted, but this value is diminished by the correctness and relevance issues.
- **Total** = 4 + 5 + 7 = 16

Model 2

- **Correctness** = 6
  - Includes both relevant and some fabricated or unclear relations (e.g., Triggers(InfrastructureDamage, old wartime munitions)).
- **Relevance** = 6
  - Many triples are relevant, though some irrelevant or vague triples are present.
- **Coverage** = 7
  - Covers a broad range of the context accurately but is diminished slightly by prioritizing the correctness and relevance.
- **Total** = 6 + 6 + 7 = 19

Model 3

- **Correctness** = 7
  - Most triples are accurate and correctly formed, with fewer fabricated relations.
- **Relevance** = 7
  - Majority of the triples are highly relevant to the context. Some triples are slightly incoherent but mostly pertinent.
- **Coverage** = 6
  - Good coverage of the context, though a few needed extractions were missing.
- **Total** = 7 + 7 + 6 = 20

Model 4

- **Correctness** = 6
  - Contains some inaccuracies but generally provides correctly formulated triples.
- **Relevance** = 6
  - Various triples are relevant to the context, but some are less directly related.
- **Coverage** = 5
  - Moderate coverage but some key elements are missed or less comprehensive.
- **Total** = 6 + 6 + 5 = 17

Model 5

- **Correctness** = 8
  - High accuracy in triple formation with minimal fabrications.
- **Relevance** = 8
  - Highly relevant triples, capturing the context well without introducing unrelated information.
- **Coverage** = 8
  - Extensive and accurate coverage of the context, handling the core relationships well.
- **Total** = 8 + 8 + 8 = 24

Ranking

1. **Model 5**
2. **Model 3**
3. **Model 2**
4. **Model 4**
5. **Model 1**