REFSafE: A RAG-Enabled Framework for Predictive Risk Analysis and Automated Safety Report Generation in Mission-Critical Environments

Sanjay Das*
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
dass3@ornl.gov

Ran Elgedawy*
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
elgedawyr@ornl.gov

Ethan Seefried
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
seefriedej@ornl.gov

Ryan Burchfield
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
burchfieldra@ornl.gov

Gavin Wiggins
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
wigginsg@ornl.gov

Dana Hewit
Savannah River National Laboratory
Aiken, South Carolina, USA
dana.hewit@srnl.gov

Sudarshan Srinivasan
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
srinivasans@ornl.gov

Prasanna Balaprakash
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
pbalapra@ornl.gov

Robert Patton
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
pattonrm@ornl.gov

Todd Thomas
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
thomastm@ornl.gov

Tirthankar Ghosal
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
ghosalt@ornl.gov

Abstract

Operational safety in mission-critical environments requires AI systems that are accurate, interpretable, and resistant to hallucination. We present an agentic Retrieval-Augmented Generation (RAG) framework, REFSafe, for grounded hazard analysis and automated safety report generation. The system integrates Large Language Models (LLMs) with structured operational data, historical incident repositories, policy documents, and external authoritative sources. Through iterative agentic reasoning, the framework retrieves, verifies, and synthesizes evidence prior to generation, enforcing citation-backed outputs with explicit source attribution (documents, links, and prior events) to ensure traceability and trust.

To mitigate hallucinations and unsupported claims, all risk assessments and forecasts are constrained to retrieved evidence, with confidence signals derived from retrieval relevance and source consistency. A transparent pipeline enables subject matter experts (SMEs) to validate predictions, and provide structured feedback, forming a continuous performance calibration loop. Preliminary deployment demonstrates improved reliability in hazard detection and safety/vulnerability report generation. This work advances trustworthy, evidence-grounded AI for predictive safety intelligence in mission-critical operations.

*Both authors contributed equally to this research.

1 Introduction

Operational safety at mission-critical and high-risk facilities demands proactive, rigorous, and continuously adaptive hazard identification. Even minor gaps in assessment can lead to catastrophic consequences, including loss of critical infrastructure, environmental damage, and human casualties. Environments such as nuclear facilities, high-voltage power laboratories, and chemical research sites inherently operate at the edge of technical and operational risk due to hazardous materials, extreme energy densities, and tightly coupled system dependencies. Despite established safety programs and regulatory controls, incidents persist—often attributable to incomplete hazard assessments, fragmented safety information distributed across heterogeneous systems, and limited awareness of historical failure patterns.

Traditional safety analysis workflows—such as hazard identification, risk indemnification assessment, safety control evaluation, and root cause analysis—remain largely manual, expert-driven, and policy-search intensive. These processes are time-consuming, often requiring days or weeks, and are susceptible to cognitive bias and oversight of emerging or non-obvious risks. While Natural Language Processing (NLP) techniques have been applied to safety documentation, including HAZOP reports (Feng et al., 2021), chemical accident databases (Single et al., 2020), and incident knowledge graphs (Wang et al., 2022), these approaches typically address isolated subtasks rather than providing integrated, end-to-end safety decision support. Even recent benchmarking efforts such as Lab-Safety Bench (Zhou et al., 2025) demonstrate that state-of-the-art

LLMs fail to exceed 70% accuracy in laboratory hazard identification, underscoring the limitations of standalone LLM reasoning in safety-critical contexts.

Recent advances in LLMs have demonstrated strong capabilities in extracting insights from unstructured safety narratives. For example, Smetana et al. (Smetana et al., 2024) applied GPT-3.5 to OSHA highway construction accident reports by clustering incident narratives using SBERT embeddings and prompting the model to summarize root causes, successfully identifying major accident categories and contributing factors. This approach revealed major accident categories (e.g. heat-related, struck-by) and their contributing factors, demonstrating that LLMs can augment traditional statistics with insights useful for prevention. Similarly, Baek et al. (Baek et al., 2025) propose a RAG-based framework where an LLM generates safety guidance by retrieving and referencing similar accident cases from a database. Their "automated safety risk guidance" model uses retrieval to ground the LLM's advice in actual past incidents. Other studies compare LLMs to human analysts in safety-risk tasks. For instance, Esposito and Palagiano (Esposito et al., 2024) evaluated fine-tuned and RAG-augmented LLMs on mission-critical security risk analysis. They found that LLMs were faster and more "actionable" than experts, and that RAG-assisted models had the lowest hallucination rates and uncovered hidden risks most comprehensively. In essence, fine-tuned LLMs gave more accurate answers while RAG models covered a broader scope of potential hazards. Some work examines LLMs on domain-specific safety narratives. Mumtarin et al. (Mumtarin et al., 2023) compared ChatGPT, Bard, and GPT-4 on U.S. crash reports: on simple yes/no questions (e.g. "Work-zone involved?"), the models agreed >85% of the time, but on complex queries (e.g. collision type) agreement fell to 35%. This suggests LLMs can extract clear facts (work-zone involvement: 96% consensus) but struggle with nuanced inferences (collision manner: 35% consensus). Taken together, these studies indicate that LLM/NLP tools can enhance text-based safety analyses (e.g. clustering narratives, extracting causal factors) and that retrieval-augmented techniques in particular help ground LLM outputs in actual safety data. However, these efforts remain fragmented and do not constitute a unified, grounded framework for predictive hazard intelligence for operational decision support.

The paradigm of RAG has fundamentally changed how LLMs interact with external knowledge, moving beyond their pre-trained parameters to incorporate real-time, domain-specific information (Lewis et al., 2020). While RAG has seen broad application in improving factual accuracy and reducing hallucinations in general knowledge-intensive NLP tasks, its specialized application in safety-critical domains represents a crucial area of ongoing research and development. Traditional RAG systems, by their nature, enhance an LLM's ability to provide more accurate and relevant responses by retrieving pertinent documents from a knowledge base. While there is a growing body of work on using AI for incident analysis (as discussed in the previous section) and general safety recommendations, the integration of RAG specifically to synthesize actionable safety insights from vast, unstructured historical incident reports is a significant advancement. This involves not just retrieving information, but semantically understanding complex safety narratives, identifying subtle precursors, and generating precise, context-aware recommendations that align with established safety protocols.

In this work, we introduce a unified, RAG-based framework, REFSafE, for predictive hazard identification in mission-critical environments. Our system integrates advanced embedding models, retrieval re-ranking, structured operational data, historical incident repositories, regulatory policies, and external sources within an iterative reasoning architecture. This fusion of advanced information retrieval and LLM-driven reasoning shifts safety management from a reactive, retrospective process toward a predictive and anticipatory paradigm. By operationalizing lessons learned at scale and transforming fragmented historical data into actionable foresight, the proposed approach enables earlier hazard detection, informed work planning, and systematic prevention of high-consequence incidents. In doing so, it advances the development of transparent, evidence-grounded AI systems capable of meeting the reliability demands of mission-critical safety operations.

Our contributions are as follows:

- We present the REFSafE system, a novel, multi-agent RAG pipeline for proactive hazard forecasting in high-risk operational environments.
- We propose a smart RAG system for intelligent retrieval of relevant incidents data from a vector database.
- We integrate Standards-Based Management System (SBMS) and external control policies for providing appropriate mitigation approaches for identified critical hazards.
- REFSafE generates comprehensive vulnerability reports combining all the relevant past events, critical missing hazards and controls towards presenting an overall risk profile of the corresponding work plan.

2 Historical Incident Repository

Past incidents are not merely records of failure—they are encoded lessons in system vulnerability, human factors, and control breakdowns. Yet in many mission-critical environments, this knowledge remains fragmented across reports, logs, and policy archives, limiting its operational value. A centralized, structured repository transforms dispersed safety narratives into actionable intelligence, enabling grounded risk reasoning, pattern discovery, and hazard forecasting.

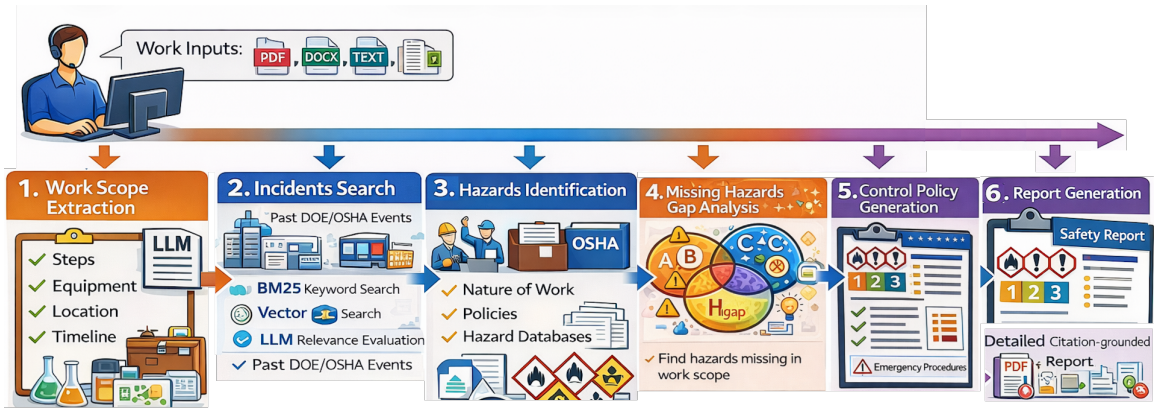

Figure 1: REFSafE system architecture. 1. Researcher uploads/inputs a work plan/order; 2. System queries multiple incidents databases in parallel for incidents 3. Identifies relevant hazards to the work scope and the past events. 4. Analyzes missing hazards gap in work plan, 5. Identifies specific controls and policies and 6. produces a comprehensive safety report.

2.1 Dataset

Our corpus combines four safety and incident reporting datasets from multiple DOE institutions (*e.g.* national laboratories), totaling over **65,000** documents spanning over three decades (1990–2025) of safety-related reporting. These sources include structured reports, narrative-style lessons-learned records, occupational injury and illness reports, and localized site-specific reports. Table 1 showcases example documents from each of the corresponding datasets.

2.1.1 ORPS

The Occurrence Reporting and Processing System (ORPS) dataset is provided by the United States Department of Energy (DOE) and documents any event that may affect public or DOE worker health and safety, the environment, national security, DOE's safeguards and security interests, the functioning of DOE facilities, or the Department's reputation (U.S. Department of Energy, 2003).

We sourced a total of 24,431 documents from the ORPS database, covering reports submitted between 1990 and 2024. Each document consists of two main components: `text` and `metadata`. The `text` field is further structured into three subfields: `title`, `summary`, and `keywords`. The metadata includes attributes such as occurrence date, facility name, reporting level, reporting criteria, process and system tags, outcome categories, and contractor information.

2.1.2 OPEXShare

The OPEXShare dataset originates from the DOE Operating Experience (OPEX) platform, which facilitates the sharing of lessons learned and best practices across DOE sites and contractors (Joseph, 2025). Each document includes a `text` field composed of a `title` and a `body`, typically describing the context, contributing factors, and corrective actions associated with a safety-related event. In total, we sourced 8,753 documents from the OPEXShare portal. These records serve as informal, narrative-style supplements to more structured reporting systems such as ORPS and CAIRS.

2.1.3 CAIRS

The Computerized Accident/Incident Reporting System (CAIRS) is maintained by the U.S. DOE and contains reports of occupational injuries and illnesses that occur during operations conducted by DOE or its contractors (Fielding, 1983). Each CAIRS document includes a `text` field, which summarizes the incident, and a rich set of metadata fields such as the date of occurrence, location, organization, contractor, injury type, severity (e.g., lost workdays), and relevant system or process tags. CAIRS provided the largest number of documents in our dataset, contributing a total of 31,109 separate reports.

2.1.4 ORNL

The ORNL dataset consists of internal safety and incident reports sourced from Oak Ridge National Laboratory (ORNL). These documents detail laboratory-specific occurrences such as procedural deviations, and equipment failures. Each entry includes a `text` field—typically a narrative description of the event, and `metadata` fields such as site, outcome tags, keywords, and lessons learned. In total, we collected 814 documents from ORNL's internal reporting system. While more localized in scope than DOE-wide systems, these reports offer valuable insight into site-specific operational hazards and institutional safety practices.

2.2 Dataset Statistics

With all four datasets combined into a single corpus, the final dataset contained 65,107 documents. Each document includes metadata such as event name, date, location, a text summary, and a full body text, among other attributes. The summaries had an average length of 231.63 ± 361.36 words, with a median of 73 words and a maximum of 3,476 words. The full body texts

Table 1: Representative Event Documents from Each Dataset.

Dataset	Textual Document Summary (Truncated)	Selected Metadata
ORPS	**Title:** Failure Of Tank 48's Purge Flow Gauge **Summary:** After the facility had entered a Limiting Condition of Operation (LCO) so that a surveillance could be performed, site personnel conducted a calibration of the Tank 48 Purge Exhaust Flow Gauge and determined that it would not hold pressure. Site maintenance personnel suspect that the gauge's diaphragm failed. A work request was initiated to replace the failed gauge, a Safety Significant Component, and restore compliance with the facility... **Keywords:** 11B - Emergency Management System Failure, 12E - Equipment Degradation/Failure	**Facility:** H Tank Farm (Nuclear Waste Operations/Disposal) **PSO:** Environmental Management **Significance:** Minor Impact **Reporting Level:** Low **Outcome Tags:** Equipment/Structural/Property Damage
OPEXShare	**Title:** Prolonged Repairs Complicate Excavation Hazard Control **Body:** Short-term controls for temporary material hazards may be inadequate for longer repair jobs and should be reevaluated periodically as a job stretches out. The excavated soil was placed on a tarp and, because the job took longer than expected, minor amounts of contamination leached out from under the tarp cover and out beyond the posted radiological control area....	**Report:** Prolonged Repairs Complicate Excavation Hazard Control **Site:** Fluor Hanford **Entity:** DOE Company/Contractor **PSO:** Environmental Management **Date:** 2000-01-05 **Outcome Tags:** Environmental Release (Hazmat, Rad, Water, etc.)
CAIRS	The staff member fell on ice while performing snow removal. He sought evaluation and treatment from the site occupational health care provider. Low back pain s/p fall. Revision 1/15/04: Sacroiliac strain. Cold pack administered....	**Title:** HANDLER/LABORER/HELPER experienced STRAIN to his/her LOWER BACK resulting in 69 lost workdays. **Site:** Pacific Northwest National Lab **Outcome Tags:** Injury, Illness, Medical Treatment; Equipment Damage
ORNL	On September 30, 2013, employees initiated a work activity, under an approved work package and two LO/TO Permits, to replace a condensate tank in Building 3047. Affected employees and the Service Supervisor attached their locks to the lock boxes. After the lock-out of the steam supply, the condensate system was drained and the drain valve left open and work activities commenced. On October 15, 2013, safety staff initiated a review of the LO/TO permits and identified several administrative proceeded...	**Lessons Learned:** None **Date:** 2023-01-28 **Outcome Tags:** Operational Safety Vulnerability, Illness, Med Treatments, or Fatalities Impacts Or Unknown

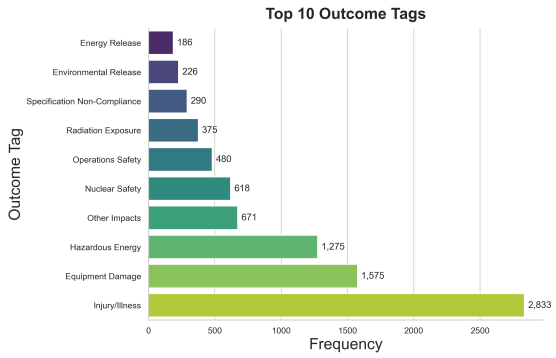

Figure 2: Frequency distribution of the top 10 outcome tags in the dataset. Energy Release (n = 186) and Environmental Release (n = 226) were the least frequent, while Injury and Illness (n = 2,833) was the most prevalent.

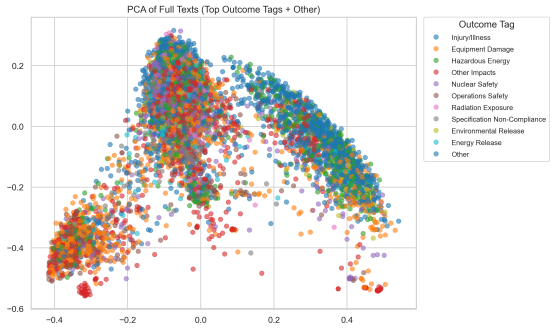

Figure 3: Two-dimensional PCA projection of TF–IDF representations of full-text documents. Points are colored by the top 10 most frequent outcome tags (shortened for clarity), with all remaining categories grouped under Other. The visualization shows clear separation between certain categories, indicating distinctive vocabulary patterns, while others overlap, reflecting shared terminology.

averaged $1,677.22 \pm 3,246.98$ words, with a median of 678 words and a maximum of 63,337 words.

To better understand the distribution of event outcomes, we examined the "outcome tags" associated with each document. Figure 2 presents the ten most frequent outcome tags, with Injury/Illness as the most common and Energy Release and Environmental Release among the least frequent. To examine the semantic organization of the corpus, Figure 3 presents a two-dimensional Principal Component Analysis (PCA) projection of the TF–IDF vector representations derived from the full-text documents. The visualization reveals that several outcome categories form well-separated clusters, indicating distinctive vocabulary and contextual patterns, while other categories exhibit partial overlap due to shared terminology across outcome types. Such insights into the corpus structure enable the analysis to prioritize the critical outcome categories (*e.g.* nuclear safety), thereby guiding the vulnerability assessment toward high-impact cases.

3 System Architecture

The REFSafe workflow is visualized in Figure 1. The system architecture as depicted in Figure 4 follows a microservices design utilizing, FastAPI for the backend server with an exposed API and a StreamLit for the

frontend, which connects to the backend.

3.1 Work Scope Extraction

Work plans or work orders (PDF or free text) are first homogenized via a Python-based pre-processing pipeline. Assume that the raw document be denoted as D_{raw}. A document normalization function $\mathcal{T} : D_{raw} \rightarrow D_{json}$ transforms heterogeneous formats (PDF, DOCX, TXT) into a standardized JSON schema:

$$D_{json} = \{\mathcal{S}, \mathcal{E}, \mathcal{L}, \mathcal{T}, \mathcal{M}\}$$

where $\mathcal{S}$ represents ordered work steps, $\mathcal{E}$ equipment/PPE, $\mathcal{L}$ location metadata, $\mathcal{T}$ timeline/duration constraints, and $\mathcal{M}$ contextual modifiers (environmental, operational conditions).

A schema-constrained LLM (*e.g.* Llama-3.1-8B-Instruct) parser $\mathcal{P}_{LLM}$ is then applied to extract specific work-scope information:

$$(\mathcal{S}, \mathcal{E}, \mathcal{L}, \mathcal{T}) = \mathcal{P}_{LLM}(D_{json})$$

Each work step $s_i \in \mathcal{S}$ is decomposed into atomic, modular actions:

$$\mathcal{S} = \{s_1, s_2, ..., s_n\}, \quad s_i = (a_i, e_i, l_i, t_i)$$

where a_i is the action, e_i the associated equipment, l_i the spatial context, and t_i the temporal constraint.

The Summarization Agent performs this step and the structured output representation enables systematic downstream hazard inference and retrieval.

3.2 Incidents Search

The incident retrieval module searches a rich database of 65,107 reports across DOE repositories using a multi-stage retrieval pipeline (by Smart Retrieval Agent). In the first stage, two independent retrieval channels operate in parallel to maximize recall. A lexical search (R_{BM25}) method captures keyword-based similarity, while a vector search (R_{vec}) identifies semantic similarity using domain-specific embeddings (*e.g.* SFR-Embedding-Mistral). The union of results from both channels forms a broad candidate set of potentially relevant incidents.

Stage 1: Dual Retrieval

$$R_{BM25} = \text{BM25}(q), \quad R_{vec} = \text{VectorSearch}(q)$$

The candidate set is:

$$R_{cand} = R_{BM25} \cup R_{vec}$$

Stage 2: Cross-Encoder Re-ranking

In the second stage, a cross-encoder model performs pairwise relevance evaluation between the work scope representation and each candidate incident. The cross-encoder computes pairwise similarity:

$$S_{ce}(q, r_i)$$

Top-k (k=30) proceed forward. This reranking step evaluates procedural, and contextual alignment, allowing the system to retain only the most relevant candidates.

Stage 3: LLM Relevance Evaluation

In this stage, an LLM (*e.g.* Llama-3.1-8B-Instruct or GPT-4o API (Application Programming Interface) call) evaluates each candidate incident across multiple qualitative dimensions, including relevance (S_{rel}), faithfulness (S_{faith}) to the original report, and contextual quality (S_{ctx}). Each candidate incident r_i is scored on:

$$S_{rel}, \quad S_{faith}, \quad S_{ctx} \in [0, 1]$$

The final ranking combines normalized retrieval (30%), reranking (40%) scores with LLM evaluation scores (40%):

$$S_{final} = 0.3 \cdot \hat{S}_{retrieval} + 0.4 \cdot \hat{S}_{rerank} + 0.3 \cdot \frac{1}{3}(S_{rel} + S_{faith} + S_{ctx})$$

where $\hat{S}_{retrieval}$ is normalized retrieval score and $\hat{S}_{rerank}$ is normalized rerank score. These scores are combined with retrieval signals to produce a final relevance score. Top-5 incidents are returned.

3.3 Hazards Identification

Hazard identification is implemented as a multi-path inference mechanism designed to maximize recall and semantic coverage.

3.3.1 Deterministic rule-based extraction

First, explicitly stated hazards are extracted directly from the structured work scope. These include hazards clearly mentioned in the work plan, such as chemical exposure, fall risk, electrical hazards, or confined space entry. A static hazard identification engine $\mathcal{H}_{plan}$ maps structured elements in D_{json} to hazards using predefined mappings:

$$A = \mathcal{H}_{plan}(D_{json})$$

3.3.2 LLM-driven contextual hazard inference

Second, a contextual inference process uses a large language model to identify latent hazards implied by the procedural context. A prompted reasoning model $\mathcal{H}_{LLM}$ generates hazards conditioned on work scope:

$$B = \mathcal{H}_{LLM}(\mathcal{S}, \mathcal{E}, \mathcal{L}, \mathcal{T})$$

These are hazards not explicitly stated but logically associated with the work activities, equipment, location conditions, or sequencing.

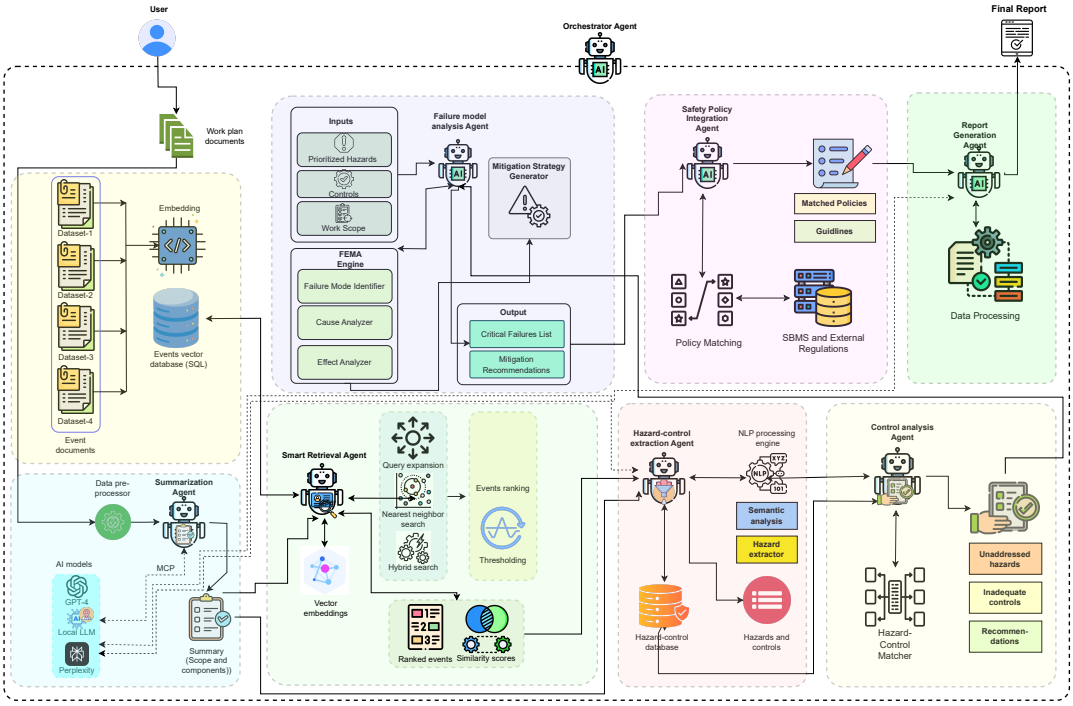

Figure 4: System architecture.

3.3.3 Hazard inference from Past Incidents

Third, hazards are derived from retrieved historical incidents. By analyzing causal factors and lessons learned in past incidents, the system identifies additional hazard patterns relevant to the current work scope.

$$C = \mathcal{H}_{Past}(\mathcal{I}_{\text{top-5}})$$

3.4 Missing Hazards Gap Analysis

The Missing Hazards Gap Analysis stage identifies hazards that were not declared in the original work plan but are historically or contextually associated with similar work. Towards this, a dedicated *Failure Model Analysis Agent* is employed with a domain-specific FEMA engine comprising: Failure Mode Identifier, Cause Analyzer and Effect Analyzer. This subsystem synthesizes a *Critical Failures List*, *causing hazard* and corresponding *Mitigation Recommendations*, which are forwarded for further analysis and policy alignment.

The final inferred hazard space is formed by combining explicitly stated hazards A, incident-derived hazards B, and contextually inferred hazards C. The union produces a comprehensive inferred hazard set:

$$H_{Inc} = A \cup B \cup C$$

The missing hazard set is computed as:

$$H_{gap} = H_{inc} \setminus H_{plan}$$

This set represents hazards historically associated with similar work but not explicitly documented in the current work plan. Such gaps are critical because workers unaware of these hazards may not receive appropriate training, implement required controls.

3.5 Control Policy Identification

For each hazard $h_j \in H_{Inc}$, control retrieval follows two parallel channels.

$$C_j = C_j^{internal} \cup C_j^{external}$$

The first channel retrieves institutional standards ($C_j^{internal}$) from indexed internal Standards-based Management System (SBMS). A cross-encoder similarity model ranks relevant control policy documents based on their alignment with the identified hazard. This, internal controls are retrieved:

$$S_{policy}(q, d_k) = \text{CrossEncoder}(q, d_k)$$

where q is the hazard query and d_k is a candidate SBMS policy document.

The second channel retrieves external best-practice guidance ($C_j^{external}$) through web-based search. This includes regulatory references, industry standards, and publicly available safety recommendations. The combination of internal and external sources ensures comprehensive hazard mitigation coverage.

3.6 Safety Report Generation

All outputs from the previous modules are consolidated into a structured safety report. The report integrates the extracted structured work scope S, explicit and inferred hazards H_{Inc}, retrieved historical incidents R_{top}, lessons learned L, gap analysis results H_{gap}, and recommended control policies C from both internal and external sources. Emergency response procedures and AI-generated safety commentary are also included.

$$\mathcal{R} = \{\mathcal{S}, H_{Inc}, R_{top}, \text{L}, H_{gap}, C\}$$

The final report is a structured and professionally formatted document that can be exported as a PDF artifact. It is suitable for pre-job briefings, regulatory documentation, supervisory review, institutional record keeping, and operational trust.

4 Evaluation and Results

This section details the choice of embedding models and the evaluation results of the different operations of the tool such as events retrieval and vulnerability report generation.

4.1 Platform and Tools

The proposed framework was implemented and evaluated on a system equipped with an NVIDIA A100 80GB GPU. The agent-based architecture was developed in Python using the PydanticAI framework, with retrieval and indexing components implemented via chromadb and LlamaIndex packages. The backend service was deployed using FastAPI with Uvicorn for asynchronous serving, while the interactive user interface was built using Streamlit. For semantic retrieval, the system employs the embedding model "SFR-Embedding-Mistral" and performs cross-encoder reranking using "cross-encoder/ms-marco-MiniLM-L6-v2". We employ the OpenAI API with the GPT-4o model to achieve state-of-the-art performance.

4.2 Evaluation of Embedding Models

To identify the optimal embedding model for REFSafe, we evaluated three high-performing embeddings based on their Hugging Face LLM leaderboard rankings (Face, 2025): SFR-Embedding-Mistral (Meng et al., 2024), OpenAI text-embedding-3-large (OpenAI, 2024), and INF-Retriever-v1 (Yang et al., 2025).

4.2.1 Experiment Setup

We randomly selected 15 ORPS documents and generated 100 QA pairs through manual and AI-assisted annotation as reference sets. Performance metrics included:

1. **Answer Correctness:** Factual agreement with reference answers (RAGAS (Es et al., 2024))

2. **Average Query Time:** Mean latency for retrieval.

4.2.2 Results

Table 2 shows INF-Retriever-v1 achieved the best balance with 68.1% correctness and 22.7s query time. SFR-Embedding-Mistral had 67.1% correctness with the fastest time (19.2s), OpenAI's model had lowest correctness (60.1%) at 20.1s. Based on these results, we selected SFR-Embedding-Mistral for subsequent experiments as it provides balance between correctness and average query time.

Table 2: Answer correctness and average query time for each embedding model.

Model	Correctness (%)	Avg Query Time (s)
SFR-Embedding-Mistral	67.1	19.2
INF-Retriever-v1	68.1	22.7
OpenAI text-embedding-3-large	60.1	20.1

4.3 Retrieval Performance Evaluation

To assess retrieval accuracy without ground-truth datasets, we applied the pooled judgment method (Sparck Jones and Van Rijsbergen, 1975; Voorhees, 2000; Tonon et al., 2015), which builds relevance assessments by merging outputs from multiple retrieval variants (Sanderson and Zobel, 2005). This approach, widely used in large-scale evaluations such as TREC (Harman, 1995; Arguello et al., 2023), is suitable for specialized domains where exhaustive labeling is infeasible.

We compared six retrieval variants: (1) *current_best* – our hybrid system using LLM-generated keywords, document titles, and CrossEncoder reranking; (2) *keywords_only*; (3) *pure_rag* – semantic similarity search; (4) *title_only*; (5) *rule_keywords* – TF-IDF and NER-based extraction; and (6) *extended_keywords* – hybrid with 10 keywords. For five test work plans, each variant retrieved the top 10 documents; up to 25 unique results per query were manually annotated on a three-point relevance scale (0–2). We employ the following metrics:

1. **Precision@5 (P@5)**: Quantifies the proportion of retrieved documents in the top 5 positions that are relevant.

$$\text{P@5} = \frac{|\text{Relevant Documents} \cap \text{Retrieved Top 5}|}{5} \tag{1}$$

2. **Recall@5 (R@5)**: Measures the fraction of the total relevant document set captured within the first 5 results.

$$\text{R@5} = \frac{|\text{Relevant Documents} \cap \text{Retrieved Top 5}|}{|\text{Total Relevant Documents}|} \tag{2}$$

3. **F1-Score@5 (F1@5)**: Provides the harmonic mean of Precision@5 and Recall@5, offering a balanced measure of retrieval performance at a narrow rank.

$$\text{F1@5} = 2 \cdot \frac{\text{P@5} \cdot \text{R@5}}{\text{P@5} + \text{R@5}} \tag{3}$$

As shown in Table 3, the hybrid system achieved the best results (F1@5 = 0.384 $\pm$ 0.080). The *title_only* variant performed closely (F1@5 = 0.369), highlighting the discriminative value of work plan titles.

4.4 Generated Report Evaluation

We assessed report quality using an LLM-as-Judge framework, following prior work showing that large language models can reliably evaluate text quality (Zheng et al., 2024; Dubois et al., 2024; Kim et al., 2024).

Figure 5: REFSafe UI overview.

Table 3: Retrieval Performance Evaluation Results Across Six System Variants.

System Variant	P@5	R@5	F1@5
RAG + keywords	0.920 ± 0.160	0.243 ± 0.053	0.384 ± 0.080
Title only	0.880 ± 0.160	0.234 ± 0.058	0.369 ± 0.086
Rule + keywords	0.800 ± 0.219	0.212 ± 0.068	0.334 ± 0.104
Keywords only	0.760 ± 0.196	0.196 ± 0.042	0.311 ± 0.069
Extended keywords	0.720 ± 0.271	0.184 ± 0.065	0.293 ± 0.104
Pure RAG	0.680 ± 0.371	0.177 ± 0.101	0.281 ± 0.158

Twenty randomly selected reports were evaluated with GPT-4 as the judge model, each compared to its corresponding work plan. Evaluations covered five criteria: *clarity* (use of technical terms), *completeness* (coverage of hazards, lessons, and mitigations), *usefulness* (support for decision-making), *accuracy* (factual grounding), and *specificity* (relevance to the work plan). Each criterion was rated on a 5-point Likert scale (1 = Poor, 5 = Excellent), with both numeric scores and textual justifications.

Table 4: Mean LLM-as-Judge ratings for generated risk reports (Likert 1–5).

Dimension	Mean Rating
Clarity	4.0
Completeness	3.0
Usefulness	4.0
Accuracy	5.0
Specificity	3.0
Overall	**3.8**

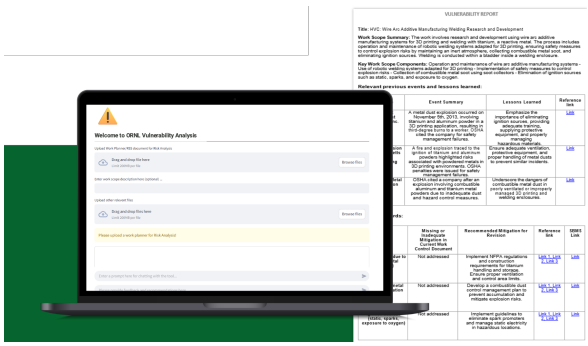

Figure 6: Tool operation example.

Accuracy received a perfect score (5.0), with GPT-4 noting that reports "consistently base hazard identifications on documented events and established safety protocols." This confirms that retrieval grounding effectively prevents unsupported claims.

Clarity* and *usefulness also scored high (4.0), reflecting precise terminology and actionable recommendations. However, lower scores for *completeness* and *specificity* (3.0 each) indicate that while major hazards were captured, some nuanced or facility-specific risks were missed. Future improvements should focus on deeper context extraction and more targeted hazard analysis.

5 Demonstration

REFSafe's user interface is designed to streamline human–AI interaction in all stages of risk analysis, focusing on transparency, traceability, and minimal analyst effort. After setup, users are directed to the main interface where submissions, results, and feedback are managed through a unified workflow, as visible in Fig. 5.

Submission Workflow: Users initiate the vulnerability analysis workflow by uploading a work plan/order with an optional scope information, and then trigger processing with a one-click action via the "Analyze vulnerabilities" button. This workflow supports both individual entries and batch uploads for multiple items.

Outputs: Upon submission, the interface analyses the work plan in the back-end via a complex RAG-enabled LLM-based risk analysis and report generation workflow, and outputs an initial safety report. Additionally the tool provides a 'Chat' option for users to interact with the tool with specific queries. Once satisfied, the users can generate a final report by clicking a "Generate final report" button. The report can be exported as a 'pdf' and downloaded.

Feedback and Review: Users can provide structured feedback through a simple form submitted via a 'Submit Feedback' button. This input is stored for audit and future model refinement.

The application's frontend provides a cohesive, context-aware experience, guiding users seamlessly from submission to analysis, report generation, feed-

back, and export, ensuring both operational efficiency and traceable process.

6 Conclusion

In this work, we proposed an end-to-end agentic workflow, REFSafe, where an SME submits a complex, high-risk work plan for automated risk assessment and hazard forecasting. REFSafe generates interpretable risk profiles and vulnerability reports, enabling experts to identify overlooked hazards and refine mitigations.

Technical Impact: REFSafe advances predictive safety in high-consequence domains by making LLM-based risk analysis *auditable*, *traceable*, and *adaptive* through agentic orchestration.

References

Jaime Arguello, Samarth Bhargav, Fernando Diaz, Evangelos Kanoulas, and Bhaskar Mitra. 2023. Overview of the trec 2023 tip-of-the-tongue track. In *The Thirty-Second Text REtrieval Conference Proceedings (TREC 2023), Gaithersburg, MD, USA, November*, pages 14–17.

Seungwon Baek, Chan Young Park, and Wooyong Jung. 2025. Automated safety risk management guidance enhanced by retrieval-augmented large language model. *Automation in Construction*, 176:106255.

Yann Dubois, Balázs Galambos, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Matteo Esposito, Francesco Palagiano, Valentina Lenarduzzi, and Davide Taibi. 2024. Beyond words: On large language models actionability in mission-critical risk analysis. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '24, page 517–527, New York, NY, USA. Association for Computing Machinery.

Hugging Face. 2025. Embedding models leaderboard. Accessed: 08-2025.

Xudong Feng, Shengnan Zhong, Qianlin Li, Laibin Zhang, and Li Ma. 2021. Application of natural language processing in HAZOP reports. *Process Safety and Environmental Protection*, 155:41–48.

JR Fielding. 1983. Computerized accident/incident reporting system (cairs) update. Technical report, EG and G Idaho, Inc., Idaho Falls (USA).

Donna Harman. 1995. Overview of the second text retrieval conference (trec-2). *Information Processing & Management*, 31(3):271–289.

Taiwo Joseph. 2025. Optimizing operational expenditure (opex) in offshore support vessel (osv) operations: A benchmarking study against international best practices.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and 1 others. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Maroa Mumtarin, Md Samiullah Chowdhury, and Jonathan Wood. 2023. Large language models in analyzing crash narratives – a comparative study of chatgpt, bard and gpt-4. *Preprint*, arXiv:2308.13563.

OpenAI. 2024. Openai text embedding 3 large. Accessed: 2025-10-28.

Mark Sanderson and Justin Zobel. 2005. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169.

Johannes I Single, Eva-Maria Schmidt, and Jörn Denecke. 2020. Knowledge acquisition from chemical accident databases using an ontology-based method and natural language processing. *Safety Science*, 129:104747.

Mason Smetana, Lucio Salles de Salles, Igor Sukharev, and Lev Khazanovich. 2024. Highway construction safety analysis using large language models. *Applied Sciences*, 14(4).

Karen Sparck Jones and Cornelis Joost Van Rijsbergen. 1975. Report on the need for and provision of an "ideal" information retrieval test collection. *British Library Research and Development Department*.

Alberto Tonon, Gianluca Demartini, and Philippe Cudré-Mauroux. 2015. Pooling-based continuous evaluation of information retrieval systems. *Information Retrieval Journal*, 18(5):445–472.

U.S. Department of Energy. 2003. *DOE M 231.1-2: Occurrence Reporting and Processing of Operations Information*. Approved: 08-19-03.

Ellen M Voorhees. 2000. Variations in relevance judgments and the measurement of retrieval effectiveness. *Information processing & management*, 36(5):697–716.

Chunling Wang, Shaojun Wu, Jianhui Guo, Qi Wang, Ruidong Zhao, and Shuicheng Tian. 2022. Using text mining to establish knowledge graph from accident/incident reports in risk assessment. *Expert Systems with Applications*, 207:117943.

Junhan Yang, Jiahe Wan, Yichen Yao, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. inf-retriever-v1 (revision 5f469d7).

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.

Yujun Zhou, Jingdong Yang, Yifan Huang, Kehan Guo, and 1 others. 2025. Benchmarking large language models on safety issues in scientific laboratories. *Nature Machine Intelligence*. ArXiv:2410.14182.