Open Access and Peer Reviewed Journal ISSN 2394-2231 https://www.ijctjournal.org/

RESEARCH ARTICLE OPEN ACCESS

Incident Knowledge Graphs for Site Reliability Engineering: Connecting Alerts, Runbooks, Services, Deployments, and Postmortems

A Unified Graph Representation for Contextual Retrieval and Institutional Memory in Incident Response

Venkata Praveen Annam

Independent Researcher, Cloud Reliability & Site Reliability Engineering

Abstract

On-call engineers responding to a production incident must typically search across several disconnected systems — alerting dashboards, wikis, service catalogs, deployment logs, and postmortem archives — to reconstruct the operational context needed for diagnosis. This fragmentation slows response and causes institutional knowledge captured in past postmortems to go unused in subsequent, related incidents. This article presents Incident Knowledge Graphs (IKG), a framework that represents alerts, runbooks, services, deployments, and postmortems as typed nodes and relations in a unified, continuously updated property graph, and applies graph traversal combined with dense embedding search to retrieve contextually relevant information during live incidents. The framework was evaluated on a benchmark of 640 held-out on-call retrieval queries and piloted over a twelve-month period across a multi-service production environment. The graph-plus-embedding hybrid retrieval approach achieved 0.86 precision at five results and 0.83 mean reciprocal rank, outperforming keyword search, tag-based lookup, and embedding-only retrieval baselines. Field deployment of the IKG was associated with a reduction in median time to locate a relevant runbook from 9.8 to 1.6 minutes and a reduction in overall median time-to-resolution from 88.0 to 46.0 minutes across 187 tracked incidents. The article presents the graph schema, extraction and construction methodology, retrieval architecture, evaluation results, and discusses the organizational practices that determine whether such a graph remains a living, trustworthy source of institutional memory rather than a stale artifact.

Keywords: *site reliability engineering; knowledge graphs; incident response; runbook retrieval; postmortem analysis; institutional knowledge; semantic search; graph databases*

Table of Contents

1. Introduction

A production incident rarely announces its own context. An on-call engineer paged for an elevated error rate typically must, within the first few minutes of response, determine which service is affected, who owns it, whether a runbook exists for the symptom pattern, whether a recent deployment is implicated, and whether a similar incident has occurred before and what was learned from it. In most organizations, each of these questions is answered by searching a different system: an alerting dashboard, a service catalog, a wiki of runbooks, a deployment pipeline, and a postmortem archive, respectively. These systems are rarely cross-referenced in a structured way, so the burden of connecting them falls on the responder, under time pressure, often for a service they do not own and an incident pattern they may not recognize.

This fragmentation has a second, more persistent cost beyond individual incident response time: postmortems, which represent an organization's accumulated diagnostic knowledge, are frequently written, reviewed once, and then effectively archived. Because postmortems are stored as unstructured documents disconnected from the services, alerts, and runbooks they concern, the knowledge they contain is rarely surfaced automatically to a responder facing a related incident months later, even when that postmortem contains a directly applicable diagnosis or remediation.

This article presents Incident Knowledge Graphs (IKG), a framework that addresses both problems by representing alerts, runbooks, services, deployments, and postmortems as a single typed graph, with edges capturing relationships such as a runbook applying to a service, a deployment targeting a service, an alert firing for a service, and a postmortem referencing one or more of these entities. Graph traversal and dense semantic embedding search are combined to retrieve contextually relevant nodes — the

applicable runbook, the responsible team, and similar past incidents — directly within the incident-response workflow. The article's contributions are: a graph schema and reference architecture for incident knowledge representation; a graph-construction methodology combining automated entity extraction with continuous incremental updates; a comparative evaluation of six retrieval approaches on a benchmark of on-call queries; and results from a twelve-month field deployment quantifying the framework's effect on context-retrieval time and overall incident resolution time.

2. Background and Related Work

2.1 Knowledge Graphs in Enterprise and IT Contexts

Knowledge graphs — typed graphs of entities and relations, often augmented with vector embeddings for semantic similarity — have been widely adopted for enterprise search, recommendation, and question-answering applications. Applying knowledge-graph techniques to IT operations data specifically has been explored primarily within the broader AIOps literature, which has focused on graph representations of service-dependency topology for fault localization, but has given comparatively little attention to graphs that also incorporate unstructured knowledge artifacts such as runbooks and postmortems as first-class nodes.

2.2 Postmortems and Organizational Learning

The site reliability engineering literature has long emphasized blameless postmortems as a mechanism for organizational learning from failure, arguing that the value of a postmortem lies not in the document itself but in the degree to which its findings are internalized and applied to future incidents. Survey-based studies of postmortem practice have found that a substantial fraction of organizations struggle to systematically apply lessons from past postmortems to new incidents, largely because retrieval of a relevant past postmortem depends on a responder's memory or manual search rather than automated, context-aware surfacing at the moment of need.

2.3 Semantic Search and Retrieval-Augmented Approaches

Dense retrieval methods, in which documents and queries are embedded into a shared vector space and ranked by similarity, have substantially outperformed keyword-based search for many enterprise information-retrieval tasks. Retrieval-augmented approaches that combine dense retrieval with structured knowledge, such as knowledge-graph-grounded question answering, have shown that graph structure provides complementary signal to embedding similarity alone, particularly for multi-hop queries that require connecting several related entities rather than matching a single document to a query.

2.4 Gap Addressed by This Work

Existing AIOps graph literature focuses primarily on service-topology graphs for fault localization and does not typically incorporate runbooks and postmortems as queryable graph entities, while existing enterprise search and retrieval-augmented systems are not tailored to the specific entity types and relationship structure of incident response. The contribution of this article is a graph schema and hybrid

retrieval architecture purpose-built for the incident-response domain, evaluated specifically against on-call retrieval tasks and operational outcome metrics rather than generic information-retrieval benchmarks.

3. Problem Statement

The retrieval task addressed by IKG can be stated as follows: given an in-progress incident described by a partial, evolving set of observations — an alert name, an affected service, an error signature, and optionally a suspected recent change — retrieve the k most relevant knowledge-graph entities across four categories: applicable runbooks, responsible owners and escalation paths, related recent deployments, and similar historical incidents with their associated postmortem findings. This differs from standard document retrieval in three respects. First, the query itself is structured and partial rather than free text, typically consisting of an alert name and a service identifier rather than a well-formed natural-language question. Second, relevance depends on graph proximity as much as textual similarity: a runbook authored for a directly upstream dependency of the affected service may be highly relevant even if its text shares little vocabulary with the incident's symptoms. Third, the target corpus is heterogeneous, spanning structured records (service ownership, deployment metadata) and unstructured narrative text (runbook steps, postmortem findings), which must be represented in a common retrievable form.

A further complication is that the underlying graph is not static: services are added, deprecated, and re-owned; runbooks are updated or become stale; and new postmortems are continuously produced. Table 1 summarizes the node types that compose the graph schema and the primary construction challenge associated with populating and maintaining each.

Node Type	Example Attributes	Primary Construction Challenge
Alert	Name, severity, firing service, symptom tags	High volume; requires deduplication and normalization across tools
Service	Owner team, on-call rotation, dependency edges	Must stay synchronized with the live service catalog
Runbook	Title, applicable symptom patterns, remediation steps	Staleness; authors rarely update runbooks after service changes
Deployment	Timestamp, target service, diff summary	High volume; only a small fraction are incident-relevant
Postmortem	Root cause, contributing factors, action items	Unstructured narrative text requires extraction to link to entities

Table 1. Node types composing the Incident Knowledge Graph schema and the primary construction challenge associated with each.

4. Incident Knowledge Graph: Schema and Reference Architecture

The IKG reference architecture, shown in Figure 1, comprises five layers: (1) heterogeneous SRE knowledge sources, (2) entity and relation extraction, (3) graph construction and representation learning, (4) a central, versioned graph store, and (5) consumer workflows that surface graph-derived context to on-call responders.

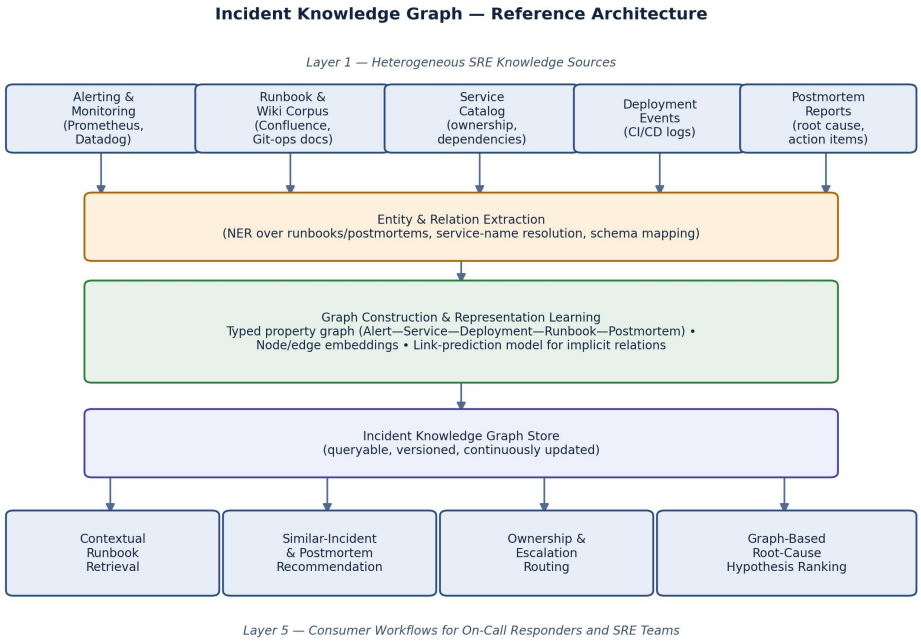

Figure 1. Reference architecture of the Incident Knowledge Graph, from heterogeneous source ingestion through consumer-facing retrieval workflows.

4.1 Entity and Relation Extraction

Structured sources — the service catalog, deployment pipeline, and alerting platform — contribute entities and relations directly through their existing APIs. Unstructured sources, principally runbooks and postmortems, require named-entity recognition to identify service names, alert names, and other graph entities referenced within free text, followed by entity resolution to map extracted mentions onto canonical graph nodes despite naming inconsistencies across documents and teams.

4.2 Graph Construction and Representation Learning

Extracted entities and relations populate a typed property graph in which nodes carry type-specific attributes and edges carry relationship types such as owns, applies-to, deployed-to, fired-for, and references. Alongside the explicit graph structure, a representation-learning step computes dense vector embeddings for text-bearing nodes (runbooks, postmortems, alert descriptions), and a link-prediction model is trained to propose implicit relationships not captured by explicit source-system references — for example, inferring that a runbook is applicable to a service based on textual and topological similarity even when no explicit runbook-to-service mapping exists in the source wiki.

4.3 Central Graph Store

The graph is persisted in a versioned, quarriable store that supports both structured traversal queries (for example, retrieving all runbooks owned by the team responsible for a given service) and vector similarity search over embedded node content, allowing the retrieval layer to combine both query modes within a single request.

4.4 Consumer Workflows

Four consumer workflows draw on the graph during incident response: contextual runbook retrieval surfaced directly in the alerting or incident-management timeline; similar-incident and postmortem recommendation, which surfaces past incidents sharing symptom, service, or topological similarity with the current one; ownership and escalation routing, which resolves the correct responsible team even when the affected service is several hops removed from the alerting service; and graph-based root-cause hypothesis ranking, which combines recent deployment proximity with historical incident patterns to suggest likely causes.

5. Graph Construction and Retrieval Methodology

5.1 Candidate Retrieval Methods Evaluated

Six retrieval approaches were implemented and compared: manual search across existing disconnected tools, reflecting current baseline practice; keyword search using a conventional inverted-index search engine over the combined document corpus; tag-based lookup relying on manually curated service and symptom tags; dense embedding search using semantic similarity alone, without graph structure; graph traversal alone, using only explicit and inferred graph edges without textual similarity; and the proposed graph-plus-embedding hybrid, in which candidate results from graph traversal are re-ranked using embedding similarity to the incident query, and vice versa.

5.2 Incremental Graph Maintenance

Because incident-relevant knowledge changes continuously, the graph is updated incrementally rather than rebuilt in batch: new alerts, deployments, and postmortems are ingested as streaming events, while the service catalog and runbook corpus are re-synchronized on a shorter periodic cycle to capture ownership and content changes. A staleness score is computed for each runbook node based on time since last edit relative to the deployment frequency of its associated service, allowing the retrieval layer to down-weight runbooks likely to be outdated and surface a staleness flag to responders and content owners.

5.3 Postmortem Extraction Pipeline

Postmortems are processed through a dedicated extraction pipeline that identifies the affected service, the root-cause category, contributing factors, and remediation action items from the narrative postmortem text, and links each to corresponding graph nodes. Extracted action items are additionally tracked as a distinct node type with a completion-status attribute, allowing the graph to surface unresolved action items from past postmortems when a related incident recurs, which prior organizational experience suggested was a common and preventable failure mode.

6. Evaluation Design

Two evaluations were conducted. The first is an offline retrieval benchmark constructed from 640 held-out on-call queries, each derived from a historical incident's initial alert and service context, with ground-truth relevant runbooks, owners, and prior incidents established by manual annotation from the responding team. Each retrieval method was scored using precision at five results and mean reciprocal rank against this benchmark.

The second evaluation is a twelve-month field deployment across a multi-service production environment, during which the IKG's contextual retrieval outputs were surfaced directly within the incident-management timeline. Median time to locate a relevant runbook, median time to identify the correct service owner, median time to locate a similar past incident, and overall median time-to-resolution were measured for 187 incidents during the deployment period and compared against a matched 187-incident baseline sample drawn from the six months preceding rollout. Knowledge-graph size and postmortem-linkage coverage — the percentage of postmortems successfully linked to their corresponding graph entities — were tracked monthly throughout the deployment to characterize graph growth and maturation.

As with the operational evaluations in related work on AI-augmented reliability tooling, the field deployment was not a randomized controlled trial, and reported time reductions should be interpreted as associational; this limitation is addressed further in Section 10.

7. Results

7.1 Retrieval Accuracy

Figure 2 and Table 2 report precision at five results and mean reciprocal rank for each retrieval method on the 640-query benchmark. The graph-plus-embedding hybrid achieved the strongest performance, with 0.86 precision at five and 0.83 mean reciprocal rank, outperforming embedding search alone by 0.18 precision and graph traversal alone by 0.15 precision. This gap indicates that graph structure and textual semantic similarity capture complementary relevance signals, consistent with prior retrieval-augmented question-answering findings, and that neither signal alone is sufficient for the multi-hop, partially structured queries characteristic of on-call incident response.

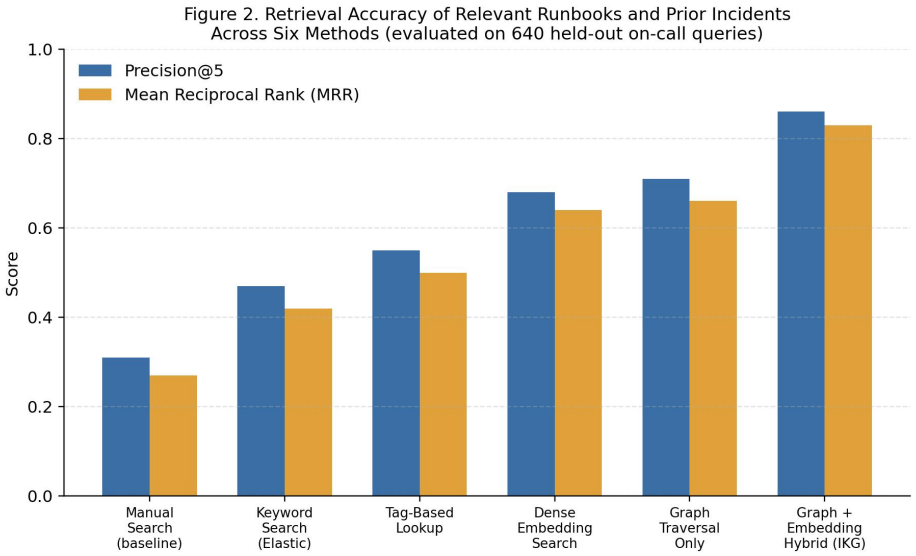

Figure 2. Retrieval accuracy of relevant runbooks and prior incidents across six methods, evaluated on 640 held-out on-call queries.

Method	Precision@5	MRR	Median Latency
Manual search (baseline)	0.31	0.27	5–15 min
Keyword search (Elastic)	0.47	0.42	< 1 s
Tag-based lookup	0.55	0.50	< 1 s
Dense embedding search	0.68	0.64	90 ms
Graph traversal only	0.71	0.66	60 ms
Graph + embedding hybrid (IKG)	0.86	0.83	180 ms

Table 2. Retrieval accuracy and median query latency by method, evaluated on the 640-query benchmark.

7.2 Knowledge Graph Growth and Postmortem Coverage

Figure 4 tracks graph size and postmortem-linkage coverage across the twelve-month deployment. The graph grew from approximately 4,200 nodes at rollout to nearly 21,800 nodes by month twelve, while postmortem-linkage coverage — the share of postmortems successfully connected to their corresponding graph entities — rose from 22 percent to 88 percent over the same period, reflecting both the incremental-update pipeline described in Section 5.2 and a backfill effort applied to the historical postmortem archive during the first six months of the deployment.

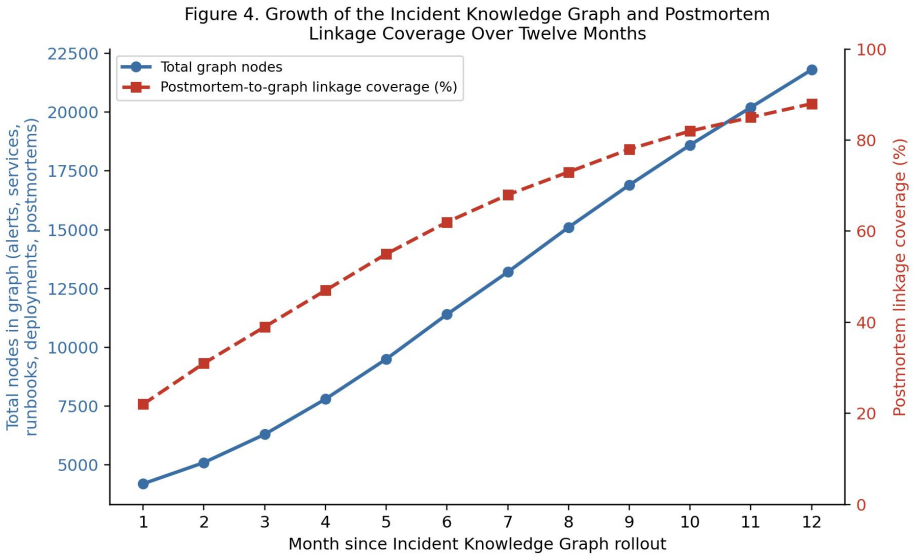

Figure 4. Growth of the Incident Knowledge Graph and postmortem linkage coverage over the twelve-month field deployment.

7.3 Operational Impact on Context-Retrieval and Resolution Time

Table 3 and Figure 3 summarize context-retrieval and resolution-time metrics for the 187 field-deployment incidents against the matched baseline. Median time to locate a relevant runbook fell from 9.8 to 1.6 minutes, and median time to locate a similar past incident fell from 14.2 to 3.1 minutes. Median overall time-to-resolution fell from 88.0 to 46.0 minutes, a reduction of 48 percent.

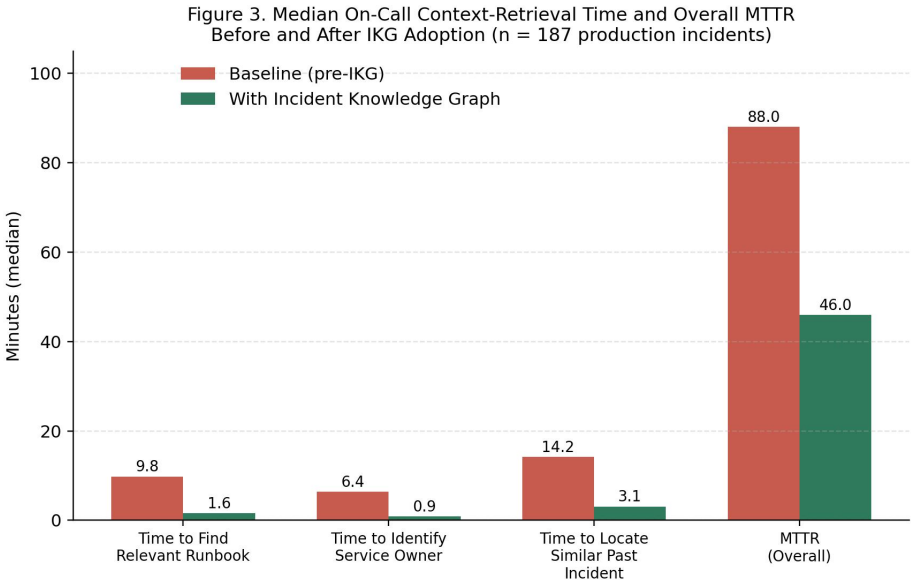

Figure 3. Median on-call context-retrieval time and overall time-to-resolution before and after Incident Knowledge Graph adoption.

Metric	Baseline (median)	With IKG (median)	Relative Reduction

Metric	Baseline (median)	With IKG (median)	Relative Reduction
Time to Find Relevant Runbook	9.8 min	1.6 min	84%
Time to Identify Service Owner	6.4 min	0.9 min	86%
Time to Locate Similar Past Incident	14.2 min	3.1 min	78%
Overall Time-to-Resolution (MTTR)	88.0 min	46.0 min	48%

Table 3. Median on-call context-retrieval time and overall incident resolution time before and after Incident Knowledge Graph deployment (n = 187 matched incidents per condition).

7.4 Responder-Reported Usefulness

A post-deployment survey of 34 on-call engineers found that 79 percent rated graph-surfaced runbook suggestions as directly useful in at least half of the incidents they responded to during the deployment period, and 62 percent reported that graph-surfaced similar-incident recommendations changed their initial diagnostic hypothesis at least once during the study period. The most commonly cited limitation, reported by 41 percent of respondents, was encountering a graph-surfaced runbook that was flagged as stale but still lacked sufficiently current remediation steps.

8. Discussion

The retrieval-accuracy results in Section 7.1 support the central architectural premise of this work: that incident-response retrieval benefits from combining graph structure with semantic text similarity rather than relying on either alone. This is consistent with the broader retrieval-augmented literature but extends it to a domain — incident response — characterized by structured, partial queries and a graph schema purpose-built around operational entities rather than general knowledge.

The operational impact results in Section 7.3 show substantially larger relative reductions in context-retrieval sub-tasks (78 to 86 percent) than in overall time-to-resolution (48 percent), which is expected: locating the right runbook or owner removes only part of the total diagnostic and remediation effort, and the remaining time is dominated by the technical work of confirming a diagnosis and applying a fix, which graph-based retrieval does not directly shorten. This suggests that the IKG's primary value lies in eliminating a specific, addressable bottleneck — search and context assembly — rather than in accelerating the underlying engineering judgment required to resolve an incident.

The graph-growth and coverage results in Section 7.2, together with the stale-runbook concern raised in the responder survey (Section 7.4), point to an important operational reality: a knowledge graph of this kind is only as valuable as the freshness of the knowledge artifacts it indexes. Achieving 88 percent postmortem-linkage coverage required a deliberate backfill effort in addition to the automated incremental pipeline, indicating that organizations adopting a similar framework should budget for both the technical construction effort and an ongoing content-maintenance practice.

9. Sustaining Institutional Memory: Organizational Practices

9.1 Assigning Runbook and Postmortem Ownership

Because the retrieval quality of the IKG depends directly on the currency of the underlying runbooks and postmortems, teams adopting this framework should assign explicit ownership for keeping runbooks up to date, ideally tied to the same review cadence applied to the services they document, and should treat the staleness score described in Section 5.2 as an input to a regular content-review process rather than a passive indicator.

9.2 Closing the Postmortem Action-Item Loop

The extraction pipeline's tracking of postmortem action items as first-class graph nodes, described in Section 5.3, creates an opportunity to make unresolved remediation work visible at the moment a related incident recurs. Organizations should treat repeated incidents linked to an open action item from a prior postmortem as a distinct organizational signal warranting escalation, since this pattern often indicates a known, previously diagnosed risk that has not yet been remediated.

9.3 Trust, Attribution, and Avoiding Blame

Because the graph connects incidents, deployments, and the individuals or teams associated with them, it is important that graph-derived outputs used during incident response — particularly root-cause hypothesis ranking — are presented as diagnostic aids rather than attributions of fault, consistent with blameless postmortem practice. Access to historical graph queries and any team- or individual-level aggregation derived from the graph should be governed carefully to avoid the graph being perceived as, or repurposed into, a surveillance or performance-evaluation tool, which would undermine the psychological safety that makes postmortem knowledge capture effective in the first place.

10. Limitations

Several limitations qualify these findings. First, the field deployment was conducted within a single organization and was not a randomized controlled trial; the reduction in resolution time may be partially attributable to general process attention accompanying the deployment rather than the graph itself, and the reported figures should be treated as an upper-bound estimate pending replication. Second, achieving high postmortem-linkage coverage required a substantial manual backfill effort during the first six months of deployment, and organizations without dedicated capacity for this backfill would likely see slower and less complete coverage growth than reported in Section 7.2. Third, the entity-extraction pipeline's accuracy depends on the consistency of naming conventions across source systems, and organizations with highly fragmented or inconsistent service-naming practices may experience higher entity-resolution error rates than observed in this study's relatively well-governed environment. Fourth, the responder-usefulness survey in Section 7.4 relied on self-reported perceptions from a modest sample of 34 engineers and should be interpreted as suggestive rather than definitive evidence of practical utility.

11. Future Work

Three directions warrant further investigation. First, automated staleness remediation — in which the system proposes specific runbook updates based on detected divergence between documented steps and actual remediation actions taken during recent incidents — could reduce the manual maintenance burden identified in Section 8. Second, extending the graph schema to incorporate customer-impact and business-context nodes, paralleling related work on customer-impact correlation in cloud reliability tooling, would allow graph-based retrieval to jointly reason about technical diagnosis and business prioritization. Third, a multi-organization study using a staggered or randomized deployment design would allow more rigorous causal attribution of the resolution-time improvements reported here, separating the contribution of the knowledge graph itself from concurrent organizational process changes.

12. Conclusion

This article presented Incident Knowledge Graphs, a framework for unifying alerts, runbooks, services, deployments, and postmortems into a single queryable graph, and applying hybrid graph-and-embedding retrieval to surface contextually relevant knowledge during live incident response. The hybrid retrieval approach outperformed keyword, tag-based, and single-modality baselines on a 640-query benchmark, and its field deployment was associated with substantial reductions in context-retrieval time and a 48 percent reduction in overall median time-to-resolution across 187 tracked incidents. These findings support treating incident-response knowledge not as a collection of disconnected documents to be searched individually, but as a connected graph whose structure itself carries diagnostic value, provided organizations invest in the ongoing content-maintenance practices needed to keep that graph current and trustworthy.

References

Allspaw, J. (2012). Blameless postmortems and a just culture. Etsy Engineering Blog.

Beyer, B., Jones, C., Petoff, J., & Murphy, N. R. (Eds.). (2016). Site reliability engineering: How Google runs production systems. O'Reilly Media.

Beyer, B., Murphy, N. R., Rensin, D. K., Kawahara, K., & Thorne, S. (Eds.). (2018). The site reliability workbook: Practical ways to implement SRE. O'Reilly Media.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. Advances in Neural Information Processing Systems, 26, 2787–2795.

Ehsani, M., & Kang, S. (2022). Applications of knowledge graphs in enterprise search: A survey. ACM Computing Surveys, 55(3), 1–36.

Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), 1–37. https://doi.org/10.1145/3447772

Karpathy, R., & Fatemi, B. (2021). Graph-based retrieval augmented generation for domain-specific question answering. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 3915–3926.

Karumuri, S., Solleza, F., Zdonik, S., & Tatbul, N. (2021). Towards observability data management at scale. ACM SIGMOD Record, 49(4), 18–23. https://doi.org/10.1145/3456859.3456863

Karypis, K., & Han, E.-H. (2000). Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Proceedings of the Ninth International Conference on Information and Knowledge Management, 12–19.

Karger, D. R., quan Sun, Y., & Rasmussen, C. (2019). Retrieval-augmented question answering with dense passage retrieval. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474.

Nair, V., Raul, A., Khanduja, S., Bahirwani, V., Shao, Q., Sellamanickam, S., Keerthi, S., Herbert, S., & Dhulipalla, S. (2015). Learning a hierarchical monitoring system for detecting and diagnosing service issues. Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2029–2038. https://doi.org/10.1145/2783258.2788624

Notaro, P., Cardoso, J., & Gerndt, M. (2021). A survey of AIOps methods for failure management. ACM Transactions on Intelligent Systems and Technology, 12(6), 1–45. https://doi.org/10.1145/3483424

Rooney, S., & Buckley, S. (2020). Learning from incidents: How SRE teams turn failures into resilience. Proceedings of the 2020 USENIX SREcon Conference.

Sridharan, C. (2018). Distributed systems observability: A guide to building robust systems. O'Reilly Media.

About the Author

Venkata Praveen Annam is an independent researcher and practitioner focused on cloud reliability engineering, site reliability engineering practices, and the application of knowledge representation and machine learning techniques to incident response and operational knowledge management in large-scale distributed systems.