FineKB: Asymmetric Multi-Centroid Retrieval for Noisy Case-to-KB Matching in Enterprise Support

Murat Kalender[1] **Aashutosh Nema**[1] **Samaksh Gulati**[1]
Jeannie M. Fitzgerald[2] **Mustafa S. Cetin**[1]
[1]Dell Technologies, Austin, Texas, USA
[2]Dell Technologies, Limerick, Ireland
{murat.kalender, aashutosh.nema, samaksh.gulati, mustafa.cetin}@dell.com

Abstract

Enterprise support retrieval requires matching noisy, multi-authored support cases to concise knowledge base (KB) articles. We present **FineKB**, an asymmetric multi-centroid retrieval framework for case-to-KB matching. FineKB summarizes historical cases offline, clusters the resulting issue summaries into per-KB centroids, and retrieves at query time using raw case text, thereby using a denoised index without per-query large language model (LLM) summarization. It further applies a lightweight reliability-aware re-ranking step based on validation-derived centroid precision. On 15,000 enterprise support cases spanning 188 KB articles, FineKB achieves 66.86% Recall@3 and 77.34% Recall@5, outperforming KB-content indexing, single-centroid representations, and a supervised fine-tuned encoder baseline, with a statistically significant Recall@5 gain over a strong case-text dense retrieval baseline. Re-ranking contributes +1.47 Recall@3 with negligible inference overhead. We release source code, anonymized vector embeddings, and cluster metadata for reproducible evaluation.[1]

1 Introduction

Enterprise support systems process large volumes of troubleshooting cases annually, and effective resolution depends on matching incoming issues to relevant knowledge base (KB) articles. A support case is a customer-reported technical incident that records the initial problem description, product and system context, diagnostic logs, troubleshooting updates, and resolution notes accumulated during the support workflow. Support cases are verbose, multi-authored texts that accumulate redundant details and terminology drift as engineers collaborate (Faheem et al., 2025; He et al., 2021; Marcuzzo et al., 2022). KB articles, by contrast, are concise technical documents optimized for clarity and reusability. This structural mismatch, compounded by lexical variation in how problems are described (Ko et al., 2006), creates a semantic gap that conventional retrieval methods struggle to bridge. As Figure 1 illustrates, a multi-paragraph case with diagnostic logs and shifting terminology must be matched to a short, precise KB solution. Enterprise KB collections also evolve continuously as articles are added, revised, or retired; retrieval approaches that require encoder retraining on each update introduce substantial maintenance overhead in production support environments.

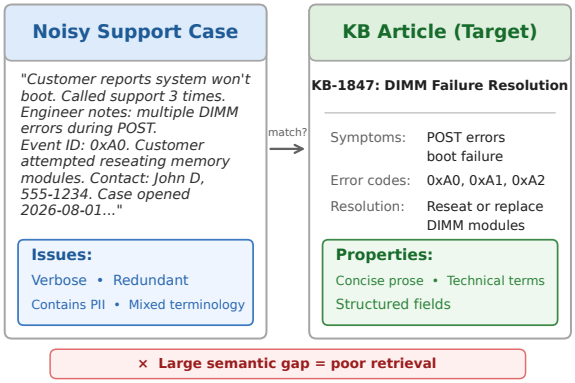


Figure 1: Semantic mismatch between noisy support cases and KB articles.

Standard retrieval approaches fail to bridge this semantic gap. Keyword methods such as BM25 and TF-IDF are sensitive to vocabulary mismatch, while dense retrievers trained on open-domain corpora struggle with domain-specific terminology and noisy multi-author case descriptions (Zhao et al., 2024). Moreover, directly indexing KB article text ignores a key source of retrieval evidence: the historical cases previously resolved by each KB article.

We present **FineKB**, an asymmetric multi-centroid retrieval framework with reliability-aware re-ranking for noisy case-to-KB matching. FineKB

[1]https://github.com/ai-research-0084/finekb

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 22164–22176
October 24-29, 2026 ©2026 Association for Computational Linguistics

summarizes historical cases offline into concise issue statements designed to retain technical signals such as error codes and affected components. It then clusters these summaries by KB article to construct per-KB multi-centroid representations of the issue patterns for which each KB has been useful. At inference time, FineKB embeds the raw incoming case and retrieves against this summary-based centroid index, avoiding per-query LLM summarization while preserving lexical and configuration cues from the raw case text. Reliability-aware re-ranking then reorders candidates using validation-derived per-centroid precision scores without additional model calls.

Evaluated on 15,000 enterprise support cases spanning 188 KB articles, FineKB outperforms KB-content indexing, single-centroid representations, and a supervised fine-tuned encoder baseline, and is competitive with or exceeds a strong case-text dense retrieval baseline. The full FineKB system achieves statistically significant gains over the supervised fine-tuned baseline on both primary metrics, though the re-ranking contribution alone does not reach significance at our test set size.

Our contributions are:

- We formulate noisy case-to-KB matching as asymmetric retrieval between raw cases and an offline summary-based, per-KB multi-centroid index, instantiated as FineKB.
- We show that clustering offline-generated case summaries into KB-specific centroids outperforms KB-content indexing, single-centroid representations, and a supervised fine-tuned encoder baseline, and is competitive with or exceeds a strong case-text dense retrieval baseline. We further show that multi-centroid indexing remains competitive even without LLM summarization.
- We introduce a lightweight reliability-aware re-ranking step using validation-derived centroid precision, improving Recall@3 with negligible inference overhead.
- We release **FineKB-Vectors**, anonymized embeddings and cluster metadata for reproducible evaluation.

2 Related Work

Enterprise KB retrieval. Enterprise support retrieval requires matching noisy, multi-authored case descriptions to concise KB articles. Lexical methods such as BM25 (Robertson and Zaragoza, 2009) and TF-IDF (Salton and Buckley, 1988) remain strong baselines but are sensitive to vocabulary mismatch. Dense retrieval methods based on Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), Sentence Transformers (Reimers and Gurevych, 2019), and Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) improve semantic matching but typically assume that indexed documents are themselves the retrieval targets. In case-to-KB matching, useful retrieval evidence may come from historical cases previously resolved by a KB article rather than from the KB article text alone. Recent work on enterprise incident and KB systems (Faheem et al., 2025; He et al., 2021; Zhao et al., 2024) and structured KB organization such as AutoKB (Sahay et al., 2025) addresses domain-specific retrieval challenges. FineKB is complementary: rather than restructuring KB content, it structures the historical case space associated with each KB article.

Multi-vector and multi-prototype retrieval. Multi-prototype embeddings (Reisinger and Mooney, 2010) and neural multi-vector retrieval models such as ColBERT and ColBERTv2 (Khattab and Zaharia, 2020; Santhanam et al., 2022) represent documents using multiple vectors to preserve fine-grained matching signals. FineKB shares this motivation but differs in granularity: rather than storing token vectors for every document, it clusters LLM-generated historical case summaries into a compact, variable-size set of KB-specific centroids, each intended to represent a distinct issue pattern for that KB article.

LLM-generated intermediates for retrieval. LLMs have been used to generate intermediate text that reduces query-document mismatch. HyDE (Gao et al., 2023) generates a hypothetical document at query time; Query2Doc (Wang et al., 2023) expands queries with pseudo-documents; and document expansion methods such as doc2query and PAQ (Nogueira et al., 2019; Lewis et al., 2021) enrich indexable content. Abstractive summarization models (Lewis et al., 2020; Raffel et al., 2020) can compress verbose inputs into retrieval-friendly representations. FineKB differs in that summaries are produced offline for historical cases and clustered into KB-specific centroids. At inference time, raw case text queries this summary-based index directly, avoiding query-time generation while preserving

lexical and configuration cues that summarization may discard.

Confidence-based re-ranking. Query performance prediction and confidence estimation have been used to detect unreliable retrieval results and adjust ranking accordingly (Zhou and Croft, 2007; Cao et al., 2008; Arabzadeh et al., 2022). FineKB computes per-centroid empirical precision on held-out validation data and uses these reliability scores to reorder candidates at inference time, adding no model calls beyond the base retrieval step.

3 Methodology

FineKB addresses the noisy case-to-KB matching problem through asymmetric multi-centroid indexing combined with reliability-aware re-ranking. The framework separates offline preprocessing from fast online inference, a design motivated by the real-time latency constraints of enterprise support workflows.

As illustrated in Figure 2, FineKB operates in two phases. The **offline indexing phase** constructs per-KB multi-centroid representations by clustering LLM-generated case summaries and computes per-centroid reliability scores on the validation set. The **online retrieval phase** performs asymmetric query matching followed by reliability-aware re-ranking, requiring only a single encoder forward pass and vector search per query.

3.1 Stratified Data Curation

FineKB requires structured case-KB associations and balanced coverage of issue types to construct robust multi-centroid representations. We transform noisy historical support logs into retrieval-ready data through two operations: consolidating case-KB associations and stratified sampling across issue categories.

Case-KB Association. Historical enterprise support logs contain free-form case descriptions initially provided by customers and later expanded by support agents with troubleshooting details. Each resolved case is linked to one KB article by support agents during resolution, providing weak supervision signals for retrieval. We consolidate these many-to-one case-KB relationships into structured pairs that define ground truth for retrieval evaluation and the grouping structure for per-KB clustering. While our study uses agent-linked associations, domains lacking such labels can employ alternative weak supervision strategies such as heuristic matching, retrieval-based pseudo-labeling, or synthetic annotations.

Stratified Sampling Across Issue Types. Historical support logs exhibit significant class imbalance: common issues dominate while rare but critical failures are underrepresented. To ensure robust evaluation and diverse cluster coverage, we stratify cases into 15 issue categories aligned with KB organization (Battery, CPU, Disk, Memory, Network, Software, etc.) and apply balanced sampling to construct train/validation/test splits (80/10/10) that preserve representation of both frequent and rare problem types. For domains lacking explicit issue-type labels, embedding-based clustering can be used to derive categories prior to sampling.

3.2 LLM-Based Issue Summarization

The issue summarizer converts verbose, multi-authored case descriptions into concise, terminology-aligned problem statements. These summaries serve as the foundation for multi-centroid indexing (Section 3.3) and can optionally provide concise issue descriptions to support agents during active troubleshooting.

Summarization Model and Prompt Design. We employ LLaMA-3.3-70B-Instruct (Llama Team, AI @ Meta, 2024) with a constrained prompt template designed to extract technical signals while discarding extraneous content. The prompt instructs the model to produce a single-sentence summary that identifies the affected product or model, component, and symptom, retaining exact error messages, codes, and event IDs, while explicitly excluding resolutions, troubleshooting steps, and personally identifiable information such as customer names, contact details, and case identifiers. The full prompt template and an illustrative input-output example are provided in Appendix A.

Role in FineKB Architecture. In the offline phase, summaries are generated for all historical cases, grouped by resolved KB article, embedded, and clustered to construct per-KB multi-centroid representations that capture heterogeneous sub-problems. In the online phase, the summarizer can optionally generate real-time summaries for display to support agents, but the core retrieval path operates asymmetrically: raw case text queries the summary-based index directly (Section 3.3), avoiding query-time summarization latency. This design

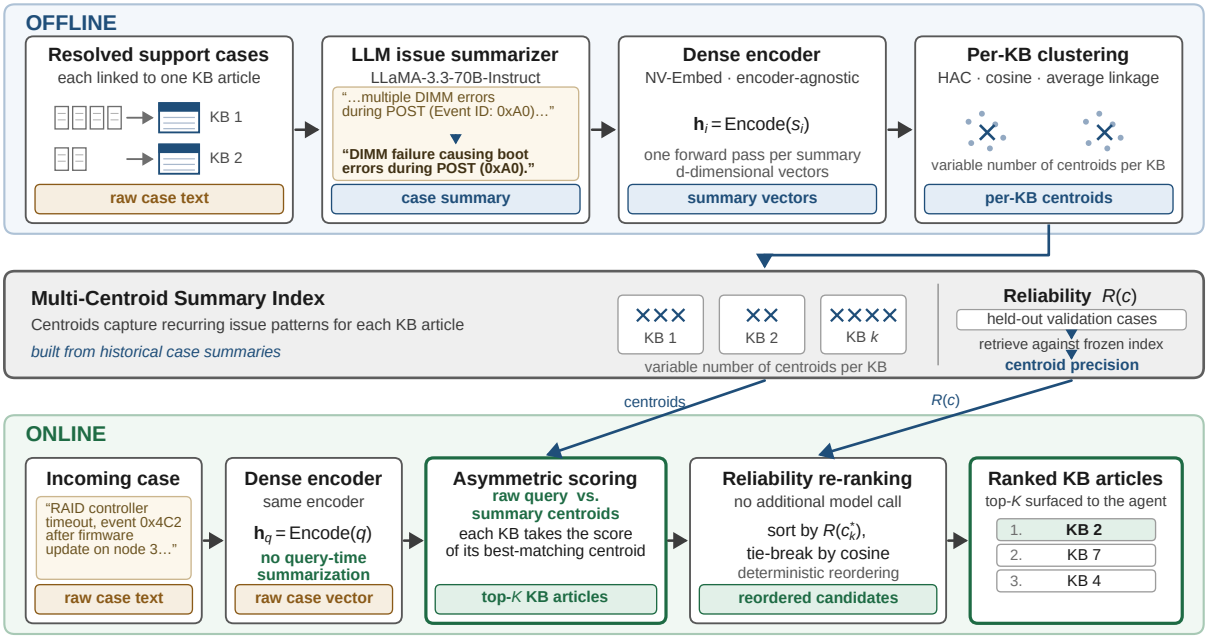


Figure 2: **FineKB architecture. Offline:** Historical case logs are processed by an LLM issue summarizer, grouped by resolved KB article, embedded, and clustered to construct a per-KB multi-centroid summary index. Centroid reliability scores are estimated from retrieval outcomes on held-out validation cases against the frozen index. **Online:** Raw case text is embedded and matched against the summary-based index via asymmetric retrieval. Retrieved candidates are reordered using the validation-derived reliability scores, requiring no additional model calls at query time.

ensures that LLM invocation occurs only once per historical case during index construction, while online retrieval remains fast through asymmetric querying.

3.3 Multi-Centroid Asymmetric Retrieval

Standard dense retrieval collapses all cases associated with a KB article into a single vector representation, obscuring the heterogeneous nature of real-world issue descriptions. For example, a KB article addressing disk failures may be linked to cases involving disk health monitoring errors, physical drive failures, controller timeouts, and array degradation, each representing a distinct sub-problem requiring different diagnostic paths. FineKB addresses this heterogeneity through per-KB multi-centroid indexing combined with asymmetric retrieval: historical case summaries are clustered to construct multiple centroid representations per KB article, while raw case text queries this summary-based index directly at inference time. Figure 3 illustrates this retrieval mechanism.

Per-KB Clustering of Historical Case Summaries. For each KB article $k \in \mathcal{K}$, we collect its associated historical case summaries $\{s_1, \ldots, s_m\}$ generated via Section 3.2 and encode them using a dense encoder:

$$\{\mathbf{h}_1, \ldots, \mathbf{h}_m\} = \text{Encode}(\{s_1, \ldots, s_m\}) \subset \mathbb{R}^d.$$

We use NV-Embed (Lee et al., 2025) for its strong semantic performance and optimized inference speed, though the framework is encoder-agnostic. We apply hierarchical agglomerative clustering (HAC) with cosine distance and average linkage to partition the embeddings into coherent sub-problem clusters. The number of clusters per KB is determined heuristically as $M_k = \lfloor n_k / s_{\min} \rfloor$, capped at $s_{\max}$, where n_k is the number of training cases for KB k, $s_{\min} = 3$ is the minimum target cluster size, and $s_{\max} = 50$ is an upper bound on clusters per KB. KB articles with $n_k \leq s_{\min}$ are assigned a single cluster. After clustering, any cluster with fewer than $s_{\min}$ cases is removed; KB articles whose all clusters are removed fall back to a single centroid over all their training cases. We use HAC rather than k-means because average-linkage clustering does not assume spherical clusters and can better accommodate uneven sub-problem distributions within a KB. In our ablations, HAC outperforms k-means by +1.05 Recall@3 under the same cluster-count heuristic (Section 4.5).

Each cluster $C_{k,j}$ is represented by its mean em-

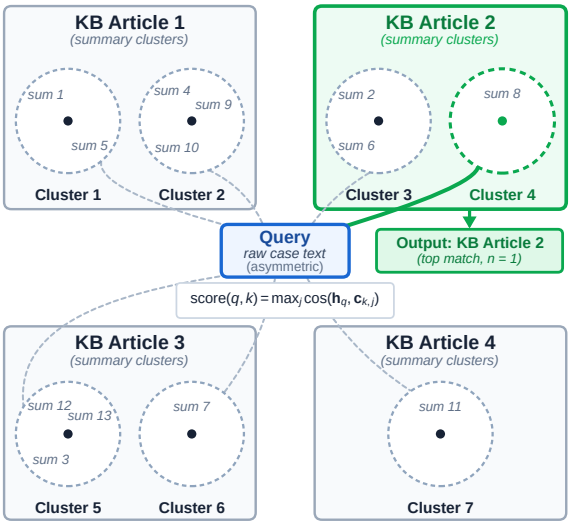


Figure 3: Multi-centroid asymmetric retrieval. A raw case query is scored against summary-based centroids across all KB articles. Each KB is assigned the score of its closest centroid; KB Article 2 is returned as the top match ($n = 1$). Dots (●) mark per-cluster centroids; labels (*sum i*) denote LLM-generated case summary embeddings.

bedding:

$$\mathbf{c}_{k,j} = \frac{1}{|C_{k,j}|} \sum_{\mathbf{h}_i \in C_{k,j}} \mathbf{h}_i,$$

yielding a multi-centroid representation $\{\mathbf{c}_{k,1}, \ldots, \mathbf{c}_{k,M_k}\}$ for KB article k.

KB-Wise versus Global Clustering. Clustering is performed on a per-KB basis rather than globally across all cases. Global clustering risks merging cases from different KB articles into shared clusters, diluting the supervision signal and producing ambiguous centroids that conflate distinct KB articles. KB-wise clustering preserves case-KB associations and allows each KB to model its specific heterogeneity independently: a frequently referenced KB may yield many centroids capturing diverse sub-problems, while a specialized KB may yield few centroids reflecting a narrower scope.

Index Construction and Scale. We construct multi-centroid indexes from both case-text embeddings and case-summary embeddings for comparison. The summary index is more compact because denoised summaries exhibit lower intra-KB variance and form denser, more semantically coherent clusters, suggesting that LLM-based summarization reduces representational variance for index construction (Table 1). For baseline comparisons, we also construct a KB-content index by directly embedding KB article text, yielding single-vector representations per KB article.

Asymmetric Retrieval. At inference time, a new case description q is embedded directly as raw text:

$$\mathbf{h}_q = \text{Encode}(q).$$

We score q against all centroids in the summary-based multi-centroid index and assign KB k the score:

$$\text{score}(q, k) = \max_{j=1,\ldots,M_k} \cos(\mathbf{h}_q, \mathbf{c}_{k,j}),$$

where the maximum over centroids selects the best-matching sub-problem cluster within KB k. This asymmetric design, querying with raw case text against a summary-based index, offers three advantages. First, raw case embedding requires only a single encoder forward pass with no query-time LLM invocation. Second, the summary-based index provides semantically aligned, denoised centroids that better capture issue structure than raw case text would. Third, raw case queries preserve idiosyncratic lexical cues such as error codes, product identifiers, and configuration details that may be discarded during summarization. All expensive operations, including LLM summarization and clustering, are performed once offline during index construction.

3.4 Reliability-Aware Re-ranking

While asymmetric multi-centroid retrieval produces strong initial rankings, centroid quality varies across KB articles depending on the density and consistency of their associated training cases. FineKB augments retrieval scores with per-centroid reliability scores derived from validation performance, promoting candidates from historically precise clusters without additional model calls at inference time.

Cluster Reliability. We compute a data-driven reliability score for each centroid using the validation split. Let $\mathcal{Q}_c = \{q' \in \mathcal{D}_{\text{val}} : \text{top1}(q') = c\}$ be the set of validation queries whose top-1 retrieved centroid is c, and let $y(q')$ denote the gold KB label for query q'. The reliability score is the empirical precision of centroid c over those queries:

$$R(c) = \frac{|\{q' \in \mathcal{Q}_c : \text{KB}(c) = y(q')\}|}{|\mathcal{Q}_c|},$$

where $\mathrm{KB}(c)$ is the KB article associated with centroid c. $R(c)$ is fixed before test-time evaluation and is not informed by any test labels. This offline computation adds no latency to the online retrieval path.

Re-ranking Procedure. After asymmetric retrieval returns the top-K KB candidates, each candidate KB article k is re-scored using lexicographic ordering over its best-matching centroid. Candidates are sorted first by $R(c_k^*)$, then by $\cos(\mathbf{h}_q, \mathbf{c}_k^*)$ as a tiebreaker, where $\mathbf{c}_k^* = \arg\max_{\mathbf{c}_{k,j}} \cos(\mathbf{h}_q, \mathbf{c}_{k,j})$ is the highest-scoring centroid vector for KB k, and c_k^* denotes its centroid identifier. This ordering promotes candidates from centroids that have demonstrated consistent precision on in-domain validation queries, without expanding the candidate set or invoking any additional model.

4 Experiments

4.1 Experimental Setup

Dataset and task. We evaluate on 15,000 English-language cases from a single enterprise IT support environment, balanced across 15 issue types: Battery, Contract, CPU, Disk, Fan, GPU, Information, Memory, Network, No Power, Power Supply, RAID, Remote Access, Software, and Storage Controller, with 1,000 cases per type. Each case contains a raw problem description initially provided by customers and later expanded by support agents. Case lengths are right-skewed (median 21 words, mean 42.1, maximum 2,620), while generated summaries are consistently concise (median 7 words, maximum 22). Across the 188 KB articles, the number of associated training cases has a median of 36 (range: 10–609). Each case is paired with one ground-truth KB article selected during resolution. We use an 80/10/10 train/validation/test split stratified by issue type. All clustering and centroid construction use only the training split; validation data are used for hyperparameter selection and reliability scoring; and test data are held out for final evaluation. A retrieved KB article is considered correct if it matches the case's ground-truth KB article.

Baselines and FineKB variants. We compare FineKB against lexical, dense, re-ranking, and generated-query baselines across two indexing strategies.

KB-content baselines index KB article text directly: BM25 uses BM25S (Lù, 2024); dense retrieval uses NV-Embed embeddings with FAISS (Facebook AI Similarity Search) IVF-Flat (Lee et al., 2025; Douze et al., 2025); HyDE (Gao et al., 2023) generates a hypothetical document at query time using GPT-5.1 (OpenAI, 2025) before retrieving over the KB-content index; and Dense (BGE-M3, supervised) (Chen et al., 2024) fine-tunes BGE-M3 on the 12,000 labeled (case, KB) training pairs using in-batch negatives and BM25-mined hard negatives with `MultipleNegativesRankingLoss`, indexing KB article text with the resulting domain-adapted encoder.

Case-text baselines index historical case text directly: BM25 and NV-Embed dense retrievers, BGE-M3 (Chen et al., 2024), Jina-ColBERT-v2 (Jha et al., 2024) (zero-shot), a general-purpose late-interaction model evaluated without domain adaptation, and a cross-encoder re-ranker (`cross-encoder/ms-marco-MiniLM-L6-v2`; Reimers and Gurevych 2019; Bajaj et al. 2016) that re-ranks top-K NV-Embed Case Text candidates against KB article text.

FineKB variants index LLM-generated issue summaries from LLaMA-3.3-70B-Instruct: the symmetric variant queries the summary index with test-case summaries; the asymmetric variant queries with raw case text. The full system, FineKB, adds reliability-aware re-ranking over the asymmetric multi-centroid index.

Metrics and evaluation protocol. We report Recall@3 and Recall@5 as primary accuracy metrics. Recall@3 is the primary metric, reflecting the production setting in which the system surfaces up to three candidate KB articles to support agents. All methods are evaluated under the same retrieval infrastructure and candidate KB universe, with embeddings precomputed offline for all dense methods. Experiments were conducted on a single NVIDIA A100 80GB GPU with batch size 32 for encoding, using FAISS IVF-Flat with `nprobe=64`. Statistical significance is assessed via paired bootstrap resampling (10,000 iterations). Due to confidentiality constraints, raw enterprise case text and KB content cannot be released; we instead release vectorized representations, indexing pipelines, retrieval code, and evaluation scripts at the repository linked in the abstract.

Table 1: Index topology statistics. k **per KB** reports min/median/mean/max clusters per KB article.

Index Source	# KBs	Clusters	k per KB
Case Text	188	790	1 / 2 / 4.20 / 38
Case Summary	188	462	1 / 1 / 2.46 / 24

Computational and maintenance cost. The dominant additional cost relative to standard dense indexing is offline summarization: the 12,000 training cases are summarized once for index construction, and new cases can be processed incrementally. On the reported hardware, clustering and centroid construction for all 188 KB articles take 3.5 seconds, and reliability-score computation over the validation split takes under 0.1 seconds. Online retrieval requires one encoder pass, one FAISS search, and deterministic re-ranking, with no LLM call. Updates are localized per KB; re-clustering the largest KB takes 0.36 seconds and requires no encoder retraining.

4.2 Cluster Index Analysis

Table 1 compares the topology of raw case-text and summary-based cluster indexes. The summary index uses 42% fewer centroids than the case-text index (462 vs. 790) and reduces the mean number of centroids per KB from 4.20 to 2.46. Mean intra-cluster cosine similarity is also higher for summary-based clusters than for raw case-text clusters (0.878 vs. 0.847), indicating that summarization produces a more compact and internally coherent index. The full cluster-size distribution is reported in Appendix B.

4.3 Sensitivity to Minimum Cluster Size

The minimum cluster size $s_{\min}$ controls both the target number of clusters and the filtering of low-support clusters. We select $s_{\min} = 3$ using validation data only. Figure 4 reports sensitivity on the held-out test set after this selection.

On the held-out test set, Recall@3 remains stable for $s_{\min} \in [1, 5]$ and is 65.39% at the validation-selected value $s_{\min} = 3$. The improvement over $s_{\min} = 1$ (64.83%) indicates that pruning low-support clusters can remove noisy or idiosyncratic case patterns. Beyond $s_{\min} = 3$, larger values produce progressively more compact indexes by discarding rare patterns, with only modest Recall@3 degradation.

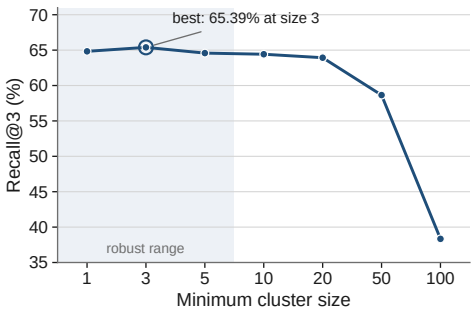


Figure 4: Test-set sensitivity of FineKB (Asymmetric) to minimum cluster size. Recall@3 remains stable for $s_{\min} \in [1, 5]$; the operating value $s_{\min} = 3$ was selected using validation data and is used in all main experiments.

4.4 Retrieval Results

Table 2 reports retrieval performance across all methods, organized by indexing strategy. Summary-based multi-centroid indexing consistently achieves the strongest accuracy, particularly when queried with raw case text.

KB-content vs. case-text indexing. Among zero-shot methods, case-text baselines substantially outperform KB-content baselines across all retrieval approaches, indicating that historical case text provides a stronger retrieval surface than KB article content alone. The gap is consistent: Dense (NV-Embed) improves from 42.73% (KB-content index) to 63.85% R@3 (case-text index) using the same encoder and query representation. HyDE achieves 44.03% R@3 on the KB-content index but requires a per-query LLM call, introducing latency incompatible with real-time support workflows, despite its marginal accuracy gain over static KB-content retrieval.

General-purpose re-ranking, late-interaction, and supervised fine-tuning. Cross-Encoder (MS MARCO) applied to Dense (NV-Embed) Case Text candidates yields 61.17% R@3, below the dense baseline without re-ranking (63.85%), suggesting that this general-purpose re-ranker does not capture the domain-specific matching signals required for enterprise case-to-KB retrieval. Jina-ColBERT-v2, evaluated zero-shot without domain adaptation, yields 56.95% R@3 despite its late-interaction architecture, falling below BM25 (60.28%) on this domain. Among zero-shot dense encoders, BGE-M3 (Chen et al., 2024) yields 57.19% R@3 compared to NV-Embed's 63.85%, justifying NV-Embed as the primary encoder.

Table 2: Retrieval performance across methods varying index source, query representation, and re-ranking strategy.

			Accuracy	
Method	**Index**	**Query**	**R@3**	**R@5**
KB-Content Baselines				
BM25	KB Content	Case Text	29.33	40.21
Dense (NV-Embed)	KB Content	Case Text	42.73	54.10
HyDE	KB Content	Hypothetical Doc	44.03	52.88
Dense (BGE-M3, supervised)	KB Content	Case Text	63.77	74.65
Case-Text Baselines				
Jina-ColBERT-v2 (zero-shot)	Case Text	Case Text	56.95	65.23
Dense (BGE-M3)	Case Text	Case Text	57.19	70.43
BM25	Case Text	Case Text	60.28	70.59
Cross-Encoder (MS MARCO)	Case Text	Case Text	61.17	74.74
Dense (NV-Embed)	Case Text	Case Text	63.85	74.74
FineKB				
– Multi-Centroid (Symm.)	Summary Clusters	Summary	59.22	71.41
– Single Centroid	Summary	Case Text	60.11	71.49
– Multi-Centroid (Symm.)	Case Text Clusters	Case Text	63.12	75.22
– Multi-Centroid (Asymm.)	Summary Clusters	Case Text	65.39	77.34
+ Reliability Re-ranking	Summary Clusters	Case Text	**66.86**	**77.34**

Fine-tuning BGE-M3 on the 12,000 labeled training pairs yields 63.77% Recall@3 and 74.65% Recall@5, a statistically significant gain over zero-shot BGE-M3 (ΔR@3=+6.58%, $p < 0.001$), showing that in-domain supervision benefits dense retrieval on this task. The full FineKB system with reliability re-ranking nonetheless outperforms the supervised fine-tuned baseline on both metrics, suggesting that the proposed representation and re-ranking design provide gains beyond in-domain encoder adaptation.

LLM-free multi-centroid operation. Although LLM summarization is an offline, one-time process over the historical case archive, its computational cost may be a constraint in resource-limited deployments. We therefore evaluate a fully LLM-free variant, Multi-Centroid (Symm.) with Case Text Clusters, which applies the same per-KB HAC clustering directly to raw case-text embeddings without any summarization step. This variant achieves 63.12% R@3 and 75.22% R@5, remaining competitive with the strongest zero-shot dense encoder (63.85%) and the supervised fine-tuned encoder (63.77%), with no statistically significant gap against either ($p > 0.05$). This result demonstrates that the multi-centroid indexing structure itself, independent of LLM summarization, provides the principal structural benefit of FineKB. LLM-generated summaries contribute an additional +2.27 R@3 gain by reducing intra-KB semantic fragmentation (Table 1), but the framework degrades gracefully when summarization is unavailable or cost-prohibitive.

Statistical significance. FineKB achieves highly significant improvements over Single Centroid on both R@3 (Δ=+5.28%, $p < 0.001$) and R@5 (Δ=+5.85%, $p < 0.001$), supporting the contribution of per-KB multi-centroid indexing. Against Dense (NV-Embed) Case Text, FineKB achieves a statistically significant R@5 gain (Δ=+2.60%, $p = 0.0016$); the R@3 improvement of +1.54% is directionally consistent but does not reach significance at our test set size. Against Dense (BGE-M3, supervised), the full FineKB system with reliability re-ranking achieves statistically significant gains on both R@3 (Δ=+3.09%, $p = 0.0238$) and R@5 (Δ=+2.69%, $p = 0.0325$), indicating that the structural contribution of per-KB multi-centroid indexing is not solely explained by the absence of in-domain supervision.

4.5 Ablation Study

Table 3 isolates the contribution of each design decision in the FineKB pipeline: index construction strategy, clustering algorithm, query representation, and reliability re-ranking.

Index construction. Multi-centroid indexing with HAC provides the largest ablation gain, yielding +5.28 R@3 over the single-centroid summary baseline. Summary-based clustering outperforms raw case-text clustering by +2.27 R@3 while using 42% fewer clusters (Table 1), suggesting that LLM-generated summaries reduce noise and yield

Table 3: Ablation results. Each section isolates one design decision while holding all others fixed at the best configuration from the preceding section. MC = Multi-Centroid; MC-Text = multi-centroid indexing applied directly to raw case-text embeddings without LLM summarization.

Configuration	R@3	R@5
Index Construction		
Single Centroid (Summary)	60.11	71.49
MC-Text (Symm.)	63.12	75.22
MC (Summary, K-Means)	64.34	74.82
MC (Summary, HAC)	65.39	77.34
Query Representation		
Symmetric (Summary → Summary)	59.22	71.41
Asymmetric (Case Text → Summary)	65.39	77.34
Reliability Re-ranking		
Without re-ranking	65.39	77.34
FineKB (with re-ranking)	**66.86**	**77.34**

a more compact centroid index. HAC outperforms k-means by +1.05 R@3, suggesting that hierarchical clustering better captures the uneven subproblem structure within KB articles.

Query representation. Asymmetric querying outperforms symmetric querying by +6.17 R@3. To examine why, two enterprise-support experts independently annotated whether summaries retained key signals in 300 held-out cases (20 per issue type; 95% raw inter-annotator agreement), following the protocol provided in Appendix C. Among cases containing each signal, summaries retained product or model identifiers in 100.0%, affected components in 97.0%, and symptoms in 93.9%, but exact error codes or event IDs in only 75.0%. Annotators found no personally identifiable information in any of the 300 evaluated summaries, corresponding to a 100% PII-free rate. This supports the asymmetric design: summarizing a query can irreversibly remove case-specific cues, whereas each index centroid aggregates recurring signals across multiple summaries and is therefore less sensitive to omissions in any individual summary.

Reliability re-ranking. Adding reliability re-ranking improves R@3 by +1.47 points with negligible inference overhead. R@5 is unchanged because re-ranking only permutes the fixed top-5 candidate set.

5 Conclusion

We presented FineKB, a retrieval framework for noisy enterprise case-to-KB matching combining asymmetric multi-centroid indexing with reliability-aware re-ranking. FineKB clusters LLM-generated issue summaries offline into per-KB multi-centroid representations, then retrieves using raw case text at query time, bridging the semantic gap between verbose case descriptions and concise KB articles without per-query LLM inference. Asymmetric multi-centroid retrieval achieves 65.39% Recall@3 and 77.34% Recall@5, outperforming KB-content indexing, single-centroid representations, and a strong case-text dense retrieval baseline. Reliability re-ranking further improves Recall@3 to 66.86% with negligible inference overhead, and the full system achieves statistically significant gains over a supervised fine-tuned encoder baseline, suggesting that the proposed design provides gains beyond in-domain encoder adaptation. We release FineKB-Vectors to support reproducible research on noisy-to-structured retrieval under enterprise distributions.

Limitations

Absence of a suitable public benchmark. No existing public benchmark we identified simultaneously exhibits the three structural properties targeted by FineKB: verbose multi-authored queries that accumulate across a case lifecycle, concise structured KB documents, and organic many-to-one case-resolution provenance with sufficient queries per KB article for multi-centroid clustering. We evaluated four candidate benchmarks, each failing on at least one critical dimension.

TechQA (Castelli et al., 2020) yields fewer than five queries per KB article, insufficient for multi-centroid clustering. NFCorpus (Boteva et al., 2016) features short, clean biomedical titles that lack the query noise and length asymmetry FineKB is designed to model. The SODD (Stack Overflow Duplicate Detection) dataset (Pasek et al., 2023) uses Stack Overflow duplicate chains, exhibiting query-to-query rather than query-to-document asymmetry. WixQA (Cohen et al., 2025) pairs enterprise user questions with Wix KB articles but does not match the target setting of FineKB: its queries are substantially shorter and cleaner than multi-authored support cases, and its per-KB query density is too low for stable multi-centroid clustering in our analysis.

LLM summarization and cold-start constraints. LLM-generated summaries may omit rare but critical details such as infrequent error codes or

edge-case configurations, potentially reducing retrieval precision when such details are the primary differentiators between similar issues. For newly created or rarely linked KB articles, FineKB falls back to KB-content dense retrieval, which lacks the case-driven normalization benefits demonstrated in our ablations.

Domain generalization. Our evaluation is conducted within a single enterprise IT support domain. Whether FineKB's structural assumptions transfer to other verticals such as healthcare, legal, or customer support remains an open question for future work.

Ethics Statement

In accordance with ACL policy, we disclose that large language models assisted with writing polish, including grammar, phrasing, and structural edits. All scientific contributions, experimental design, analyses, and claims were conceived, implemented, and verified by the human authors.

The dataset consists of confidential enterprise support cases and KB articles. To balance reproducibility with privacy, we release only precomputed vector embeddings and cluster structures, with no raw text or personally identifiable information included. All processing was conducted in a secure environment following organizational privacy policies. Released artifacts are intended for research purposes only and must not be used to reconstruct or re-identify customers or systems.

Reproducibility Statement

We provide detailed methodology (Section 3) and experimental setup (Section 4.1) to support reproduction of our results. Because raw enterprise text cannot be released under organizational confidentiality constraints, we release **FineKB-Vectors** containing: (1) NV-Embed embeddings of LLM-generated case summaries across train/validation/test splits, (2) NV-Embed embeddings of KB articles, and (3) metadata including KB identifiers, issue types, cluster assignments, and per-centroid reliability scores. These vectors enable replication of the retrieval, clustering, reranking, and ablation results in Tables 2 and 3. The repository also includes indexing pipelines, retrieval code, and hyperparameter configurations for reproducing the released-vector experiments.

References

Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal*, 25(4):365–385.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016*, pages 716–722, Padova, Italy. Springer.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250.

Vittorio Castelli, Rishav Chakravarti, Saswati Dana, Anthony Ferritto, Radu Florian, Martin Franz, Dinesh Garg, Dinesh Khandelwal, Scott McCarley, Michael McCawley, Mohamed Nasr, Lin Pan, Cezar Pendus, John Pitrelli, Saurabh Pujar, Salim Roukos, Andrzej Sakrajda, Avi Sil, Rosario Uceda-Sosa, and 2 others. 2020. The TechQA dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1278, Online. Association for Computational Linguistics.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Dvir Cohen, Lin Burg, Sviatoslav Pykhnivskyi, Hagit Gur, Stanislav Kovynov, Olga Atzmon, and Gilad Barkan. 2025. WixQA: A multi-dataset benchmark for enterprise retrieval-augmented generation. *Preprint*, arXiv:2505.08643.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss library. *IEEE Transactions on Big Data*.

Muhammad Faheem, Muhammad Awais, Aqib Iqbal, and Hasnain Zia. 2025. Enhancing IT incident management with natural language processing and predictive analytics. *International Journal of Science and Research Archive*, 15(3):224–237.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Shilin He, Pinjia He, Zhuangbin Chen, Tianyi Yang, Yuxin Su, and Michael R Lyu. 2021. A survey on automated log analysis for reliability engineering. *ACM computing surveys (CSUR)*, 54(6):1–37.

Rohan Jha, Bo Wang, Michael Günther, Georgios Mastrapas, Saba Sturua, Isabelle Mohr, Andreas Koukounas, Mohammad Kalim Akram, Nan Wang, and Han Xiao. 2024. Jina-ColBERT-v2: A general-purpose multilingual late interaction retriever. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 159–166, Miami, Florida, USA. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 39–48, New York, NY, USA. Association for Computing Machinery.

Amy J Ko, Brad A Myers, and Duen Horng Chau. 2006. A linguistic analysis of how people describe software problems. In *Visual Languages and Human-Centric Computing (VL/HCC'06)*, pages 127–134. IEEE.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. NV-Embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Llama Team, AI @ Meta. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xing Han Lù. 2024. BM25S: Orders of magnitude faster lexical search via eager sparse scoring. *arXiv preprint arXiv:2407.03618*.

Matteo Marcuzzo, Alessandro Zangari, Michele Schiavinato, Lorenzo Giudice, Andrea Gasparetto, and Andrea Albarelli. 2022. A multi-level approach for hierarchical ticket classification. In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)*, pages 201–214, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Jimmy J. Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*.

OpenAI. 2025. GPT-5.1 instant and GPT-5.1 thinking system card addendum. `https://openai.com/index/gpt-5-system-card-addendum-gpt-5-1/`. Accessed: 2026-05-15.

Jan Pasek, Jakub Sido, Miloslav Konopik, and Ondrej Prazak. 2023. MQDD: Pre-training of multimodal question duplicity detection for software engineering domain. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 824–835, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer (T5). *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10,

pages 109–117, USA. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Rishav Sahay, Arihant Jain, Purav Aggarwal, and Anoop Saladi. 2025. AutoKB: Automated creation of structured knowledge bases for domain-specific support. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 708–723, Albuquerque, New Mexico. Association for Computational Linguistics.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Trans. Inf. Syst.*, 42(4).

Yun Zhou and W Bruce Croft. 2007. Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 543–550.

A Prompt Template and Summarization Example

The following prompt template was used to generate issue summaries from historical case descriptions using LLaMA-3.3-70B-Instruct. The template enforces consistency by explicitly specifying required technical elements and prohibited non-technical content.

Prompt Template

You are a support assistant for enterprise server products.
Summarize the technical issue in one sentence.
Rules:
1) Include product/model if available.
2) Specify affected component and symptom.
3) Retain exact error messages, codes, or event IDs.
4) Exclude resolutions, troubleshooting steps, customer identifiers, names, emails, phone numbers, addresses, dates, signatures, and case IDs.

The following example illustrates the transformation from a raw case description to a generated summary.

> **Raw Case Description.** "[Phone] System reports multiple DIMM errors during POST (Event ID: 0xA0). Customer attempted reseating. Contact: John D, 555-1234. Preferred Contact Method: Email."
>
> **Generated Summary.** "DIMM failure causing boot errors during POST (Event ID: 0xA0)."

The model extracts critical technical signals, including the affected component (DIMM), symptom (boot errors), diagnostic context (POST), and error code (0xA0), while discarding customer metadata and troubleshooting actions.

B Cluster-Size Distribution

Figure 5 shows the distribution of cluster sizes, measured as the number of historical cases assigned to each centroid. The raw case-text index is heavily concentrated in micro-clusters, particularly the 0–9 case bucket, indicating that noisy multi-turn support text often fails to aggregate into stable semantic groups. The summary-based index shifts mass toward larger clusters, suggesting that LLM-generated issue summaries reduce lexical and conversational noise and produce a more compact cluster distribution. This pattern is consistent with the main-paper finding that summary-based indexing reduces fragmentation while preserving the benefits of multi-centroid retrieval.

C Human Annotation Protocol

Two enterprise-support experts independently evaluated LLaMA-3.3-70B-Instruct summaries for 300 held-out cases, with 20 cases sampled from each issue type. The annotators were internal salaried

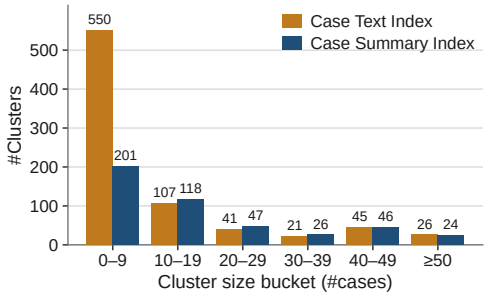


Figure 5: Distribution of cluster sizes measured as cases per centroid. The **Case Text** index is skewed toward micro-clusters, reflecting high semantic fragmentation. The **Case Summary** index yields a more balanced distribution with fewer micro-clusters and more larger clusters, indicating a more compact centroid index.

employees selected for their enterprise-support expertise and completed the evaluation as part of their regular duties, without additional task-specific compensation. Annotators were instructed to compare each summary only with its original case description and not with other summaries. For every applicable item, they assigned a binary judgment indicating whether the summary (1) preserved the correct product or model, (2) retained exact error codes, event IDs, or error messages, (3) identified the affected component, (4) captured the principal symptom or failure behavior, and (5) contained no personally identifiable information. Items absent from the original case were marked as not applicable and excluded from the corresponding retention-rate calculation. Raw inter-annotator agreement across applicable binary judgments was 95%.