SAFEINTENT: Answer-Format Query Expansion Without Answer-Prior Leakage for Private Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

LLM-based query expansion methods improve retrieval by asking a language model to generate a hypothetical answer document. On public QA benchmarks, however, such documents conflate two signals: answer-like format alignment with relevant passages and the model's parametric prior over the answer. This conflation is risky for corpus-private RAG, where the relevant entities, aliases, or facts may be unknown to the LLM. We isolate this issue with a controlled private-like transformation of public benchmarks that synchronously renames query and corpus anchors, preserving query–corpus alignment while invalidating public-world entity priors. Under this setting, free-form pseudo-document expansion can degrade dense retrieval by shifting representations away from renamed evidence. To bridge this gap, we introduce SAFEINTENT: a training-free answer-format reformatting method that does not retrieve with free-form pseudo-documents. Instead, it distills answer-free bridge terms, combines them with query-supported intent, and produces validated retrieval views for retrieval and fusion. SAFEINTENT improves both BM25 and dense retrieval on transformed queries, suggesting that private RAG benefits from separating answer-format alignment from unreliable answer priors.

1 Introduction

Retrieval-augmented generation (RAG) systems depend on retrieval to surface evidence before generation (Lewis et al., 2020). Because user queries are often short, underspecified, or phrased differently from corpus passages, retrieval systems have long used query expansion and feedback to reduce query–document mismatch (Rocchio Jr, 1971; Lavrenko and Croft, 2017; Cormack et al., 2009). Recent LLM-based methods extend this idea by asking a language model to generate a hypothetical answer document and retrieving with that generated text (Gao et al., 2023; Wang et al., 2023). This hypothetical document expansion is effective but confounded. A generated pseudo-document may help by supplying relational words, contextual phrases, and document-style wording that align the query with the corpus. It may also help because the LLM already memorizes the answer and writes answer-bearing content into the retrieval text. Public QA benchmarks reward both effects, making it difficult to distinguish useful answer-format alignment from answer-prior leakage.

This distinction matters for corpus-private RAG. We use "private" in the sense of facts unavailable to the LLM's parametric memory. Enterprise, medical, legal, scientific, and internal support corpora often contain private aliases, product names, incident IDs, evolving facts, or domain-specific entities that an LLM cannot know. In such settings, a fluent pseudo-document may introduce the wrong public-world entity, treat an unfamiliar token as a placeholder, or move a dense query representation away from the private anchor that identifies the relevant evidence. We study this failure mode using a controlled private-like transformation of Natural Questions (NQ) (Kwiatkowski et al., 2019) and SciFact (Wadden et al., 2020). The transformation synchronously renames selected anchors in both queries and corpus passages, preserving retrieval alignment while invalidating public-world entity priors. This lets us test whether retrieval-time generation still helps when public answer priors are unreliable.

Our results show that public gains from QUERY2DOC (Wang et al., 2023) do not transfer uniformly to the renamed setting. On public NQ and SciFact, answer-aware masking reveals that QUERY2DOC gains mix answer-bearing content with format or evidence-style reformulation: masking reduces gains on NQ, while remaining competitive with unmasked QUERY2DOC on SciFact. Under synchronized renaming, the effect splits by retriever: BM25 can still benefit from lexical rela-

tion terms, while free-form pseudo-documents can hurt dense retrieval, especially when unsupported generated content shifts representations away from renamed corpus evidence.

To address the failures in the private RAG setting, we propose SAFEINTENT, a training-free method that preserves answer-like retrieval structure without allowing concrete generated answers into the final retrieval text. Rather than retrieving with a free-form pseudo-document, SAFEINTENT extracts answer-free bridge terms, validates them against query-supported intent, and converts the validated intent into retrieval views: a deterministic sparse bridge view and constrained dense evidence sketches that are fused with the original query. This keeps the useful format-alignment signal while blocking query-external entities, answer guesses, and refusal-like text from entering retrieval.

Thus, our contributions are: (i) we identify answer-prior leakage as a confound in LLM-based query expansion for RAG; (ii) we introduce a synchronized renaming protocol that preserves NQ and SciFact retrieval while invalidating public-world entity priors; (iii) we show that free-form pseudo-document expansion can harm dense retrieval on renamed private-effective queries even when it remains useful for public or lexical retrieval; and (iv) we present SAFEINTENT, an answer-free reformatting method that improves both sparse and dense retrieval by separating answer-format alignment from unreliable answer priors.

2 Related Work

Query expansion and retrieval. Query expansion has long addressed lexical mismatch between short queries and relevant documents through relevance feedback, relevance models, and pseudo-relevance feedback (Rocchio Jr, 1971; Lavrenko and Croft, 2017; Abdul-Jaleel et al., 2004; Robertson and Zaragoza, 2009). Rank fusion provides a simple way to combine alternative query views (Cormack et al., 2009). Neural sparse and dense retrievers change the representation space, but retain the same need to bridge user wording and corpus wording (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Izacard et al., 2021; Formal et al., 2021).

LLM-based query expansion. Recent work uses LLMs to generate retrieval-time text. HYDE embeds hypothetical documents for zero-shot dense retrieval (Gao et al., 2023), while QUERY2DOC uses LLM-written pseudo-documents for sparse and dense query expansion (Wang et al., 2023). Follow-up works study directions such as generative relevance feedback, corpus-steered expansion, verification, and hallucination filtering (Mackie et al., 2023; Lei et al., 2024; Jia et al., 2024; Zhang et al., 2024; Wang et al., 2024; Liu and Zhang, 2025b,a; Kuo et al., 2025; Li et al., 2025b,a; Xia et al., 2025; Li et al., 2025d,c; Yao et al., 2025; Xu et al., 2025). These methods generally treat generated text as retrieval signal. Our work asks when that signal becomes unsafe because it carries public-world answer priors into retrieval.

Contamination, conflicts, and RAG robustness. Our concern is related to benchmark contamination, memorization, and knowledge conflicts in language models (Carlini et al., 2021, 2022; Yang et al., 2023; Dong et al., 2024; Li et al., 2024a; Xu et al., 2024b,a; Chen et al., 2025). Prior contamination studies often ask whether evaluation is inflated because a model has seen test data; we study a retrieval-time channel where a rewriter can inject parametric answers before evidence is retrieved. On the other hand, unlike prior RAG works that study query–knowledge relevance, conflicts between parametric memory and retrieved evidence, and robustness to counterfactual or noisy contexts (Li et al., 2024b; Wang et al., 2025; Hong et al., 2024; Lee et al., 2025; Perçin et al., 2025; Ju et al., 2026), we focus on the earlier retrieval step, where answer priors can shape the query representation itself.

Private-like and entity-sensitive retrieval. Privacy-preserving RAG studies leakage of private queries, retrieved records, or retrieval databases (Zeng et al., 2024, 2025; Cheng et al., 2025; Koga et al., 2024; Choi et al., 2025). Our setting is different: we do not study disclosure of private records, but whether public parametric priors remain useful when query and corpus facts are aligned yet no longer match the public world. Our renaming protocol is connected to counterfactual and robustness evaluations (Kaushik et al., 2019; Gardner et al., 2020; Ribeiro et al., 2020) and to entity-renaming and entity-centric retrieval studies showing that familiar names, private aliases, and entity representations can strongly affect retrieval behavior (Yan et al., 2022; Sciavolino et al., 2021; Nguyen et al., 2024; Yamada et al., 2025; Fayyaz et al., 2025). We use

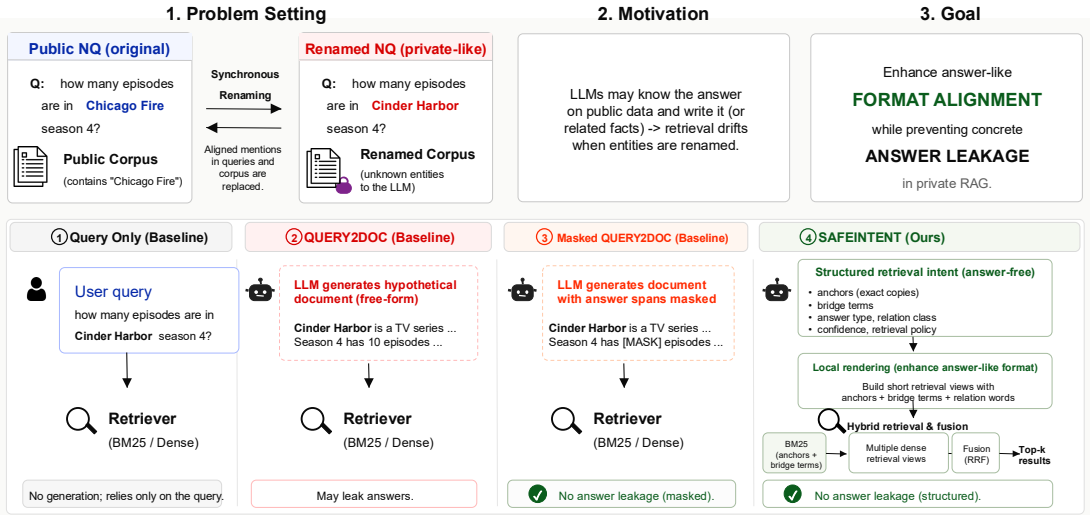

Figure 1: Overview of the private-like RAG setting and retrieval methods. Synchronous renaming preserves query-corpus alignment while making public-world answer priors unreliable. SAFEINTENT preserves answer-like format alignment by extracting structured, answer-free retrieval intent before local validation and rendering.

renaming not as an adversarial perturbation, but as a controlled way to preserve the retrieval problem while invalidating the LLM's public answer prior.

3 Problem Setup

Let $\mathcal{C} = \{d_i\}$ be a retrieval corpus, q a user query, and $R(q, \mathcal{C})$ the ranked list returned by a retriever. Query expansion introduces a rewriting function g and retrieves using either $g(q)$ or a concatenation $q \oplus g(q)$. In QUERY2DOC-style expansion, $g(q)$ is a generated pseudo-document intended to answer q.

We separate two possible sources of gain: (1) **format signal**, where the generated text resembles relevant passages and supplies relation words, context words, and document-style phrasing; and (2) **answer-prior signal**, where the generated text contains the answer or answer-adjacent facts from the LLM's parametric memory. The second signal is unreliable in corpus-private RAG, where relevant evidence may depend on names, aliases, or facts absent from the LLM's parametric memory.

Research questions. We organize the study around three questions. **RQ1**: How much of the public QUERY2DOC gain remains after answer-bearing spans are masked? **RQ2**: Does QUERY2DOC still help when corpus and query anchors are renamed consistently, so public-world priors are unreliable? **RQ3**: Can answer-like format alignment improve retrieval when concrete answer generation is disallowed?

3.1 Private-Like Renaming

To isolate this issue, we construct a renamed version of NQ and SciFact, respectively. We define a mapping ρ over selected query-corpus anchor spans and apply it consistently to both the corpus and the queries. Candidate anchors are first proposed from the query, where the LLM identifies complete replaceable spans and their coarse semantic types, such as a TV series title, person, organization, or location. The code then validates that each proposed span is an exact query substring and that the same span occurs in the aligned relevant corpus passages. If a source query contains a validated anchor span s and the relevant corpus passages also contain s, both are rewritten to $\rho(s)$. Thus, the query and the corpus remain mutually consistent, but public-world priors about s no longer apply.

We use two replacement modes to balance independence from public-world knowledge and robustness of the evaluation. **Opaque** replacements use code-like identifiers such as `Work_2J0504`. This mode provides the strongest separation from the LLM's public entity priors, since the replacement tokens cannot be resolved from pretraining knowledge. However, such identifiers can also look artificial to the LLM. In practice, the model may treat them as placeholders or private codes and produce meta responses such as asking for external search or additional context, rather than attempting an answer-like reformulation. This behavior is not the failure mode we aim to measure.

To reduce this artifact, we also construct a **Plau-**

sible replacement mode. Plausible replacements use synthetic but type-matched name-like identifiers, such as replacing the TV series title "Chicago Fire" with `Cinder Harbor`. These replacements still break the link to the original public entity, but they look more like ordinary corpus entities of the same broad type. This setting better approximates private or enterprise collections that contain real-looking internal names, project names, products, or organizations that are not known to the LLM, while reducing direct refusals caused by obviously artificial tokens.

For example, the source query "how many episodes are in Chicago Fire season 4" becomes `how many episodes are in Work_2J0504 season 4` in the opaque mode and `how many episodes are in Cinder Harbor season 4` in the plausible mode. In both modes, replacements are applied consistently to the query and all aligned corpus mentions, while relation words and numeric modifiers such as `season 4` are preserved. This keeps the retrieval target fixed while removing the reliability of the original public-world entity prior.

Figure 2 illustrates the synchronized renaming pipeline. The final evaluation uses 500 queries and 200,000 passages for NQ, and 300 queries and 5,183 documents for SciFact. In NQ, 284 of the 500 queries contain at least one validated query–corpus replacement and are treated as the *private-effective* subset, while the remaining 216 queries are kept as unchanged controls. In SciFact, 223 of the 300 queries are private-effective, and 77 are unchanged controls. See Appendix B for detailed statistics.

The two datasets differ in scale and replacement structure. NQ contains 2,042 mapping rows, with 1,794 entity replacements and 248 number replacements. SciFact contains 1,317 mapping rows, with 1,136 entity replacements and 181 value replacements. For both datasets, the query-only BM25 audit shows a 0.0000 nDCG@10 change after renaming. This indicates that the synchronized transformation preserves first-stage query-only retrieval behavior while changing the public-world assumptions available to an LLM generator.

3.2 Evaluation Question

The central question is whether a retrieval-time generator still helps when it cannot rely on public answers. If QUERY2DOC mostly contributes answer-like format alignment, a leakage-free method should be able to preserve that benefit after synchronized renaming. If its public gains depend on answer priors, we expect free-form pseudo-documents to degrade on the private-effective subset, especially for dense retrieval where fluent but wrong pseudo-documents can shift the embedding representation away from the query anchors.

4 Methods

4.1 Baselines

Query-only retrieval. The first baseline retrieves directly with the user query q. This is the target that a reformatter must improve without relying on answer generation.

QUERY2DOC. The standard LLM expansion baseline prompts the model to "write a passage that answers the given query." The generated pseudo-document is retrieved by itself and also appended to the original query. We report the expanded-query variant, which is strongest in the final summary. For BM25, the original query is repeated before appending the pseudo-document; for dense retrieval, the query and pseudo-document are joined with `[SEP]`.

MASKED-QUERY2DOC. To test how much QUERY2DOC depends on explicit answer strings, we add a query-aware masking baseline. Given the query and generated pseudo-document, an LLM masks spans that directly fill the unknown answer with typed slots such as `[PERSON]`, `[LOCATION]`, or `[NUMBER]`. This removes some explicit leakage, but it operates after the pseudo-document has already been generated. If the pseudo-document has snapped to the wrong public-world entity, masking cannot fully repair it.

We omit baselines such as HYDE because QUERY2DOC is an enhanced version of it.

4.2 SAFEINTENT: Leakage-Free Answer-Format Expansion

SAFEINTENT aims to strengthen answer-like format alignment while removing unreliable answer-prior content from retrieval. A pseudo-document generated by QUERY2DOC may contain unsupported concrete answers or public-world priors, but it can also expose useful bridge terms: relation words and evidence expressions that often appear near the answer in corpus passages. SAFEINTENT therefore does not retrieve with the full pseudo-document. Instead, it distills answer-free, relation-level bridge terms from the pseudo-document,

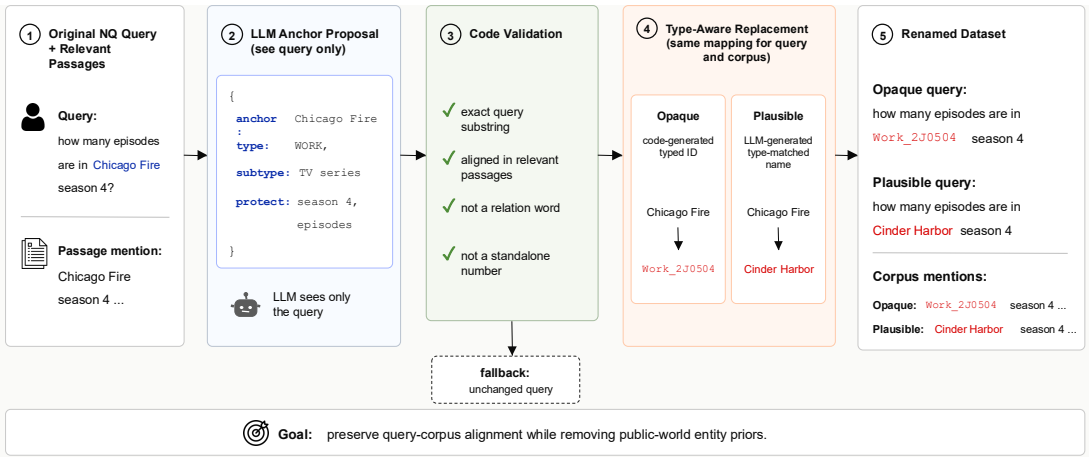

Figure 2: Private-like renaming pipeline. Starting from an original public query and its aligned relevant passages, an LLM proposes complete query anchor spans and coarse types while identifying protected relation words and modifiers. Code validation keeps only anchors that are exact query substrings, appear in relevant passages, and are not relation words or standalone numbers. Each validated anchor is then replaced consistently in the query and corpus with either an opaque typed identifier or a plausible type-matched synthetic name; queries without a validated anchor fall back to the unchanged-control split.

filters out concrete answer guesses and query-external priors, and uses the remaining terms to construct validated answer-format retrieval views.

SAFEINTENT first treats the QUERY2DOC pseudo-document as a candidate source of bridge terms. Query-external proper nouns, dates, numbers, typed answer slots, refusal-like text, and low-quality terms are removed. The remaining terms are kept only when they are generic relation or evidence expressions consistent with the query relation. These bridge terms do not represent the concrete answer; they represent how relevant passages may express the answer-bearing relation.

To assess the role of distillation, we also include an ablation called SAFEINTENT without distill. This variant keeps the same intent schema, sanitizer, renderer, and fusion procedure, but does not use the QUERY2DOC pseudo-document as a source of bridge terms. Instead, the LLM directly generates generic relation terms from the original query under the same answer-free constraints. This ablation is used only to test whether bridge terms distilled from the pseudo-document provide additional retrieval signal beyond query-only relation prediction.

Given the original query, query anchors, and distilled bridge terms, SAFEINTENT forms a constrained retrieval intent with the following fields: (1) an answer type from a closed set; (2) anchors copied exactly from the query; (3) a relation frame containing a query-supported subject, relation phrase, answer slot, and query constraints; (4) bridge terms distilled from the pseudo-document; and (5) a confidence label and retrieval policy.

A local sanitizer validates the intent before rendering any retrieval text. Anchors must be exact query substrings, the answer type must belong to the closed set, and the relation frame must be query-supported. Bridge terms must not contain query-external concrete entities, years, numbers, titles, or refusal-like language. If validation fails, the system falls back to a conservative intent supported only by the original query.

The validated intent is converted into answer-free retrieval views with different strategies for sparse and dense retrieval.

For sparse retrieval, SAFEINTENT uses a single bridge view formed by concatenating the original query, copied query anchors, and sanitized relation-level bridge terms. This preserves the query anchors while adding controlled lexical cues for BM25 matching.

For dense retrieval, SAFEINTENT renders multiple short answer-format evidence sketches. Each sketch preserves the query anchors and relation frame, keeps the answer slot unfilled, and uses sanitized bridge terms to express a possible evidence pattern. Retrieval is performed separately with the original query and each validated sketch, and the resulting rankings are combined with group-normalized weighted reciprocal-rank fusion. Let $\mathcal{V}$ denote the set of validated sketches, and let $r_q(d)$

and $r_v(d)$ denote the rank of document d under the original query and sketch v, respectively. The final dense score is

$$S(d) = \lambda \frac{1}{k + r_q(d)} + (1 - \lambda) \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{1}{k + r_v(d)}.$$

If a document does not appear in a ranking, its contribution from that ranking is set to zero. We use the standard RRF constant $k = 60$ and set $\lambda = 0.6$, so that the original query receives the largest total weight. Averaging the sketch scores before fusion prevents examples with more validated sketches from receiving additional weight. If no sketch passes validation, SAFEINTENT falls back to query-only dense retrieval. In this way, the rendered sketches provide answer-like relation structure without replacing the original query with a single fluent pseudo-document.

5 Experimental Setup

Retrievers. We use two first-stage retrievers. The sparse retriever is BM25 with standard tokenization over alphanumeric and underscore tokens. The dense retriever uses `BAAI/bge-base-en-v1.5` sentence-transformer embeddings (Xiao et al., 2023), normalized dot-product scoring, and the standard query prefix for passage search. Both retrievers return the top 20 documents.

Generator. All LLM artifacts use `GPT-5-mini` and are cached (Singh et al., 2025). QUERY2DOC uses a few-shot prompt for a passage answering the query. MASKED-QUERY2DOC uses query-aware masking over the generated passage. SAFEINTENT uses the structured intent prompt in Appendix D.

Metrics. We use nDCG@10 as the primary metric because it reflects both retrieving relevant passages and ranking them near the top. We also report MRR@10 and Recall@10 as auxiliary metrics. On renamed data, the main comparison uses the private-effective subset, where renaming actually changes the LLM prior; full-set results are reported as controls because unchanged queries can hide private-specific failures.

6 Results

6.1 Public Results

Table 1 reports public retrieval results on NQ and SciFact. QUERY2DOC improves both BM25 and dense retrieval on both datasets, showing that pseudo-document expansion provides useful retrieval signal beyond the original query. However, answer-aware masking reveals different mixtures of signals across the two benchmarks. On NQ, MASKED-QUERY2DOC consistently remains below unmasked QUERY2DOC while still improving over query-only retrieval, indicating that explicit answer-bearing content contributes substantially to the public gain. This pattern is consistent with the factoid structure of NQ, where queries often ask for a concrete entity, location, date, or number that a pseudo-document can directly insert into the retrieval text.

Dataset	Retriever	Metric	Query	Q2D	Masked Q2D
Natural Questions					
NQ	BM25	nDCG@10	0.3818	**0.6049**	0.5424
		MRR@10	0.3384	**0.5622**	0.4987
		Recall@10	0.5700	**0.8001**	0.7440
	Dense	nDCG@10	0.6868	**0.7451**	0.7178
		MRR@10	0.6388	**0.7018**	0.6723
		Recall@10	0.8997	**0.9287**	0.9143
SciFact					
SciFact	BM25	nDCG@10	0.6644	**0.7240**	0.7234
		MRR@10	0.6337	**0.6822**	0.6819
		Recall@10	0.7816	**0.8744**	**0.8744**
	Dense	nDCG@10	0.7404	0.7680	**0.7687**
		MRR@10	0.7034	**0.7289**	0.7286
		Recall@10	0.8742	0.9103	**0.9148**

Table 1: Public retrieval results. QUERY2DOC improves BM25 and dense retrieval across NQ and SciFact, while answer-aware masking reduces but does not eliminate the gains on NQ and remains competitive on SciFact.

SciFact shows a different pattern. MASKED-QUERY2DOC remains close to QUERY2DOC and slightly improves some dense metrics, suggesting that the benefit of pseudo-document expansion is less tied to concrete answer strings. Because SciFact queries are scientific claims rather than direct factoid questions, relevant abstracts are retrieved largely through terminology, relation phrasing, and evidence-style reformulation. Masking therefore removes less useful signal than in NQ and may also reduce over-specific generated content. Overall, these public results show that pseudo-document expansion mixes answer-bearing content with answer-like or evidence-style reformulation, but the relative contribution of the two signals depends on the retrieval task.

6.2 Opaque vs. Plausible Renaming

Before comparing retrieval methods, we first examine whether the two renaming modes induce different generation behavior. Specifically, we measure

Dataset	**Renaming mode**	**Meta-refusal rate**
NQ	Opaque	68.82%
NQ	Plausible	7.33%
SciFact	Opaque	54.60%
SciFact	Plausible	5.80%

Table 2: Meta-refusal rate of QUERY2DOC under private-like transformations. Plausible renaming reduces refusal behavior on both NQ and SciFact, making it better suited for testing answer-prior hallucination rather than placeholder sensitivity.

how often QUERY2DOC produces a meta-refusal instead of an answer-like pseudo-document. We define meta-refusals as responses that treat the renamed entity as a placeholder or private code, ask for the original name, request external search, or state that the model cannot identify the entity.

Table 2 shows a large gap between the two settings. Opaque identifiers trigger meta-refusals in 68.82% of cases, while plausible renaming reduces the rate to 7.33%. SciFact shows the same pattern: opaque identifiers trigger meta-refusals in 54.60% of cases, while plausible renaming reduces the rate to 5.80%. This suggests that opaque renaming often tests placeholder sensitivity rather than the failure mode we target: whether generated answer-like content can introduce harmful public-world priors for private data. Plausible renaming substantially reduces this refusal artifact while still breaking the link to the original public entity. Therefore, unless otherwise stated, the following renamed-corpus experiments focus on the plausible setting.

6.3 Private-Effective Renamed Results

We now evaluate retrieval behavior under plausible renaming. Because the full renamed sets include unchanged controls, aggregate results can obscure the effect of private-like transformations. We therefore focus on the *private-effective* subsets, where at least one validated query–corpus anchor has been replaced. This yields 284 queries for NQ and 223 queries for SciFact, directly testing whether answer-like reformulation remains useful when public-world entity priors are unreliable.

Table 3 shows that sparse retrieval remains more tolerant of pseudo-document expansion than dense retrieval. For BM25, QUERY2DOC improves over query-only retrieval on both datasets, increasing nDCG@10 from 0.3931 to 0.5362 on NQ and from 0.6560 to 0.7021 on SciFact, with similar gains in MRR@10 and Recall@10. This indicates that pseudo-documents still provide useful lexical evidence under renaming, including answer-format phrasing, relation terms, and corpus-style expressions that improve sparse matching.

Dense retrieval is more sensitive to unsupported generated content. On NQ, QUERY2DOC falls below query-only retrieval across all three metrics, reducing nDCG@10 from 0.6827 to 0.6454. This supports the failure mode targeted by our study: fluent pseudo-documents can shift dense representations toward public-world priors that are no longer faithful to the renamed corpus. SciFact shows a milder version of the same issue. QUERY2DOC provides no clear dense gain, with nDCG@10 slightly decreasing from 0.7332 to 0.7311 while MRR@10 and Recall@10 remain close to query-only. This weaker degradation is consistent with SciFact's claim-based structure, where queries already contain richer terminology and relation context than many factoid NQ questions.

Answer-aware masking partially mitigates this dense-retrieval failure. On both datasets, MASKED-QUERY2DOC improves dense retrieval over unmasked QUERY2DOC and exceeds query-only retrieval. The same pattern appears in BM25, where masking slightly improves over QUERY2DOC on both NQ and SciFact. These results suggest that removing concrete generated answers and query-external facts can reduce semantic drift while preserving useful answer-format or evidence-style structure.

The SAFEINTENT without-distill ablation further clarifies the role of distilled bridge terms. This variant is answer-free, but its relation terms are predicted directly from the query rather than distilled from the pseudo-document. It consistently underperforms MASKED-QUERY2DOC across datasets, retrievers, and metrics. This gap suggests that query-only relation terms are often too generic or mismatched with the actual wording of relevant documents, making it difficult to predict the most useful evidence expressions from the query alone. Thus, answer-oriented pseudo-document generation can expose useful corpus-style evidence expressions, but these expressions must be separated from concrete answer guesses before retrieval.

SAFEINTENT achieves the strongest results across all reported metrics for both BM25 and dense retrieval across datasets. By filtering out concrete answer guesses and query-external priors while retaining distilled bridge terms, SAFEINTENT strengthens answer-like format alignment without allowing generated answers to enter re-

Dataset	Regime	Retriever	Metric	Query	QUERY2DOC	MASKED-QUERY2DOC	SAFEINTENT w/o distill	SAFEINTENT
Natural Questions: plausible renamed private-effective subset								
NQ	Plausible	BM25	nDCG@10	0.3931	0.5362	0.5368	0.5325	**0.5737**
			MRR@10	0.3423	0.4863	0.4876	0.4744	**0.5371**
			Recall@10	0.5784	0.7161	0.7282	0.6936	**0.7683**
		Dense	nDCG@10	0.6827	0.6454	0.7126	0.7108	**0.7312**
			MRR@10	0.6393	0.6158	0.6689	0.6632	**0.6824**
			Recall@10	0.9001	0.8877	0.9137	0.9087	**0.9213**
SciFact: plausible renamed private-effective subset								
SciFact	Plausible	BM25	nDCG@10	0.6560	0.7021	0.7069	0.6818	**0.7346**
			MRR@10	0.6301	0.6608	0.6632	0.6570	**0.6897**
			Recall@10	0.7741	0.8486	0.8521	0.8315	**0.8763**
		Dense	nDCG@10	0.7332	0.7311	0.7589	0.7480	**0.7693**
			MRR@10	0.6955	0.6961	0.7204	0.7147	**0.7288**
			Recall@10	0.8660	0.8669	0.9026	0.8852	**0.9171**

Table 3: Private-effective retrieval results under plausible renaming for NQ and SciFact. Dense SAFEINTENT uses group-normalized fusion with $\lambda = 0.6$.

Dataset	Retriever	Δ nDCG@10	95% CI	p_{Holm}
NQ	BM25	$+0.0369$	$[+0.013, +0.064]$	0.011
NQ	Dense	$+0.0186$	$[+0.004, +0.036]$	0.031
SciFact	BM25	$+0.0277$	$[+0.008, +0.050]$	0.014
SciFact	Dense	$+0.0104$	$[-0.004, +0.023]$	0.139

Table 4: Paired significance results for SAFEINTENT over MASKED-QUERY2DOC on plausible-renamed private-effective subsets. Δ is the mean paired difference in nDCG@10. Confidence intervals are from paired bootstrap resampling over queries, and p_{Holm} denotes two-sided paired randomization-test p-values after Holm correction.

Fusion weight λ	Dense nDCG@10
0.5	0.7274
0.6	**0.7312**
0.7	0.7253
0.8	0.7116

Table 5: Effect of the fusion weight λ on dense SAFEINTENT retrieval under plausible renaming on the NQ private-effective subset.

trieval. The resulting retrieval views align better with renamed corpus evidence than all baselines.

We also test significance on the plausible-renamed private-effective subsets using paired query-level randomization tests with paired bootstrap confidence intervals. Since nDCG@10 is our primary metric, Table 4 reports paired query-level significance tests for SAFEINTENT vs. MASKED-QUERY2DOC. SAFEINTENT improves mean nDCG@10 in all four settings, with statistically significant gains on NQ BM25, NQ dense, and SciFact BM25 after Holm correction.

6.4 Effect of Fusion Weight

We further analyze the dense fusion weight λ, which balances the original query against the auxiliary answer-format sketch group. As shown in Table 5, performance is stable for moderate values and peaks at $\lambda = 0.6$. Giving the sketches equal weight slightly reduces performance, while assigning too much weight to the original query underuses the auxiliary answer-format signal. We therefore use $\lambda = 0.6$ in the main experiments.

7 Conclusion

LLM-based query expansion can improve retrieval by producing answer-like, passage-aligned text, but public QA gains can conflate useful format alignment with answer content injected from parametric memory. Our public results show this entanglement: answer-aware masking removes partial gains of QUERY2DOC. To test its consequence for private-like retrieval, we constructed synchronized, renamed versions of data that preserve query–corpus alignment while making public-world answer priors unreliable. In these settings, QUERY2DOC can degrade retrieval on the private-effective subset, especially for dense retrieval, where unsupported generated content can shift representations away from renamed corpus evidence.

We propose SAFEINTENT, a leakage-free answer-format expansion method that strengthens answer-like format alignment without allowing concrete generated answers to enter retrieval. SAFEINTENT improves both BM25 and dense retrieval on renamed private-effective queries across NQ and SciFact. These findings suggest that retrieval-time generation for private RAG should preserve useful answer-format alignment while separating it from unreliable answer priors.

Ethics Statement

This work studies retrieval behavior using public benchmark data and synthetic, synchronized renaming. The renamed identifiers are artificial and are not intended to represent real people or organizations. The motivation is to make private-RAG evaluation more conservative: systems should not be rewarded for injecting unsupported public-world facts into retrieval text. A deployment of leakage-free answer-like reformatting should still respect access control, logging policies, and user privacy, since query rewriting can expose sensitive query terms to an external model if implemented with a hosted API.

Limitations

Our experiments are designed to isolate a specific retrieval-time failure mode rather than to provide a complete benchmark for private RAG. We use 500 NQ queries with 200,000 passages and 300 SciFact queries with 5,183 documents, so broader datasets and larger corpora are needed to test how well the observed behavior generalizes.

The renamed NQ and SciFact benchmarks are private-like proxies rather than real private corpora. Synchronous renaming preserves query–corpus alignment while making public-world entity priors unreliable, but real private corpora may contain domain-specific terminology, evolving facts, access-control constraints, and more complex document structure that entity renaming alone does not capture.

SAFEINTENT also relies on a fixed intent schema and an LLM call at query time. This design helps prevent concrete generated answers from entering retrieval, but may require extension for domain-specific relations, multi-hop evidence, or deployment settings with strict latency, cost, or privacy constraints.

References

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. 2004. Umass at trec 2004: Novelty and hard.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and 1 others. 2025. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10091–10109.

Yihang Cheng, Lan Zhang, Junyang Wang, Mu Yuan, and Yunhao Yao. 2025. Remoterag: A privacy-preserving llm cloud rag service. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3820–3837.

Yujin Choi, Youngjoo Park, Junyoung Byun, Jaewook Lee, and Jinseong Park. 2025. Safeguarding privacy of retrieval data against membership inference attacks: Is this query too close to home? *arXiv preprint arXiv:2505.22061*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050.

Mohsen Fayyaz, Ali Modarressi, Hinrich Schuetze, and Nanyun Peng. 2025. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9136–9152.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

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.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, and 1 others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the*

Association for Computational Linguistics: EMNLP 2020, pages 1307–1323.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. 2024. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2474–2495.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Pengyue Jia, Yiding Liu, Xiangyu Zhao, Xiaopeng Li, Changying Hao, Shuaiqiang Wang, and Dawei Yin. 2024. Mill: Mutual verification with large language models for zero-shot query expansion. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2498–2518.

Li Ju, Junzhe Wang, and Qi Zhang. 2026. Faithfulness-qa: A counterfactual entity substitution dataset for training context-faithful rag models. *arXiv preprint arXiv:2604.25313*.

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.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

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*, pages 39–48.

Tatsuki Koga, Ruihan Wu, Zhiyuan Zhang, and Kamalika Chaudhuri. 2024. Privacy-preserving retrieval-augmented generation with differential privacy. *arXiv preprint arXiv:2412.04697*.

Yuan-Ching Kuo, Yi Yu, Chih-Ming Chen, and Chuan-Ju Wang. 2025. Mmlf: Multi-query multi-passage late fusion retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6587–6598.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Victor Lavrenko and W Bruce Croft. 2017. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA.

Jungyeon Lee, Kangmin Lee, and Taeuk Kim. 2025. Magic: A multi-hop and graph-based benchmark for inter-context conflicts in retrieval-augmented generation. *arXiv preprint arXiv:2507.21544*.

Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-steered query expansion with large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–401.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Hang Li, Xiao Wang, Bevan Koopman, and Guido Zuccon. 2025a. Pseudo-relevance feedback can improve zero-shot llm-based dense retrieval. *arXiv e-prints*, pages arXiv–2503.

Hang Li, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2025b. Llm-vprf: Large language model based vector pseudo relevance feedback. *arXiv preprint arXiv:2504.01448*.

Minghan Li, Xinxuan Lv, Junjie Zou, Tongna Chen, Chao Zhang, Suchao An, Ercong Nie, and Guodong Zhou. 2025c. Query expansion in the age of pre-trained and large language models: A comprehensive survey. *ACM Transactions on Information Systems*.

Rui Li, Liyang He, Qi Liu, Zheng Zhang, Heng Yu, Yuyang Ye, Linbo Zhu, and Yu Su. 2025d. Unirag: Unified query understanding method for retrieval augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14163–14178.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024a. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 528–541.

Zhuohang Li, Jiaxin Zhang, Chao Yan, Kamalika Das, Sricharan Kumar, Murat Kantarcioglu, and Bradley A Malin. 2024b. Do you know what you are talking about? characterizing query-knowledge relevance for reliable retrieval augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6130–6151.

Lingyuan Liu and Mengxiang Zhang. 2025a. Exp4fuse: A rank fusion framework for enhanced sparse retrieval using large language model-based query expansion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 163–173.

Lingyuan Liu and Mengxiang Zhang. 2025b. Golfer: Smaller lms-generated documents hallucination filter & combiner for query expansion in information retrieval. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 153–162.

Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative relevance feedback with large language models. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 2026–2031.

Thong Nguyen, Shubham Chatterjee, Sean MacAvaney, Iain Mackie, Jeff Dalton, and Andrew Yates. 2024. Dyvo: Dynamic vocabularies for learned sparse retrieval with entities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 767–783.

Sezen Perçin, Xin Su, Qutub Sha Syed, Phillip Howard, Aleksei Kuvshinov, Leo Schwinn, and Kay-Ulrich Scholl. 2025. Investigating the robustness of retrieval-augmented generation at the query level. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 439–457.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4902–4912.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*, volume 4. Now Publishers Inc.

Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. 2025. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30553–30571.

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.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, and 1 others. 2024. Searching for best practices in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17716–17736.

Yu Xia, Junda Wu, Sungchul Kim, Tong Yu, Ryan A Rossi, Haoliang Wang, and Julian McAuley. 2025. Knowledge-aware query expansion with large language models for textual and relational retrieval. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4275–4286.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Cheng Xu, Shuhao Guan, Derek Greene, M Kechadi, and 1 others. 2024a. Benchmark data contamination of large language models: A survey. *arXiv preprint arXiv:2406.04244*.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024b. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*.

Zhichao Xu, Shengyao Zhuang, Xueguang Ma, Bingsen Chen, Yijun Tian, Fengran Mo, Jie Cao, and Vivek Srikumar. 2025. Rethinking on-policy optimization for query augmentation. *arXiv preprint arXiv:2510.17139*.

Ikuya Yamada, Ryokan Ri, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. 2025. Dynamic injection of entity knowledge into dense retrievers. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16867–16879.

Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. 2022. On the robustness of reading comprehension models to entity renaming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 508–520.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*.

Sijia Yao, Pengcheng Huang, Zhenghao Liu, Yu Gu, Yukun Yan, Shi Yu, and Ge Yu. 2025. Llm-qe: Improving query expansion by aligning large language

models with ranking preferences. *arXiv e-prints*, pages arXiv–2502.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yiding Liu, Yue Xing, Han Xu, Jie Ren, Yi Chang, Shuaiqiang Wang, Dawei Yin, and 1 others. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4505–4524.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Jie Ren, Tianqi Zheng, Hanqing Lu, Han Xu, Hui Liu, Yue Xing, and Jiliang Tang. 2025. Mitigating the privacy issues in retrieval-augmented generation (rag) via pure synthetic data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24538–24569.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024. Exploring the best practices of query expansion with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1872–1883.

A The Use of Large Language Models (LLMs)

LLMs are used as experimental components to generate QUERY2DOC pseudo-documents, masking outputs, and SAFEINTENT intents or rendered sketches, as described in the method and prompt appendices. Separately, LLMs were used only for writing assistance such as proofreading and grammar polishing. All ideas, claims, methods, experimental design, results, and references were determined by the authors.

B Renamed Dataset Statistics

Table 6 summarizes the renamed dataset construction.

C Qualitative Example of SAFEINTENT Rendering

Table 7 illustrates how SAFEINTENT differs from free-form pseudo-document expansion on a plausible private-like renamed query. QUERY2DOC generates a fluent answer-like passage, but the passage includes a concrete answer and additional unsupported details that may reflect public-world priors rather than corpus evidence. In contrast, SAFEINTENT treats the pseudo-document only as a noisy source of relation-level bridge cues. It marks concrete answers and query-external details as unsafe, preserves the renamed query anchors, and produces validated answer-free retrieval views: a sparse bridge view built from the original query and safe bridge cues, and dense sketches with the answer represented by a typed slot. This example shows how SAFEINTENT retains answer-format alignment while preventing generated answer content from entering retrieval.

Dataset	Property	Value
NQ	Source dataset	Natural Questions
	Corpus size	200,000 passages
	Queries	500
	Private-effective queries	284
	Unchanged controls	216
	Replacement modes	opaque, plausible
	Mapping rows	2,042
	Entity / number replacements	1,794 / 248
	Query-only BM25 nDCG@10 delta	0.0000
SciFact	Source dataset	SciFact
	Corpus size	5,183 documents
	Queries	300
	Private-effective queries	223
	Unchanged controls	77
	Replacement modes	opaque, plausible
	Mapping rows	1317
	Entity / value replacements	1136 / 181
	Query-only BM25 nDCG@10 delta	0.0000

Table 6: Renamed dataset construction statistics for NQ and SciFact, including private-effective queries, unchanged controls, replacement counts, and query-only BM25 nDCG@10 change after renaming.

D SAFEINTENT Prompt Templates

This appendix provides the prompt templates used by SAFEINTENT. The prompts are used only to produce structured, answer-free retrieval intents. Local code parses and validates all JSON outputs before any generated text is used for retrieval.

D.1 Bridge-Cue Distillation Prompt

This prompt extracts an answer-free retrieval intent from the original query and the QUERY2DOC pseudo-document.

Prompt 1: Bridge-Cue Distillation

```
You are a leakage-free retrieval intent distiller for
      private RAG.

The query may contain plausible but renamed entities, such
      as synthetic show names, people, organizations,
      products, or places. Treat these names as real corpus
       anchors. Do not decide whether they are known
      entities. Do not answer the query from world
      knowledge.

You are given:
1. the original query;
2. a Query2Doc pseudo-document generated for that query.

Your task is to compare the query with the pseudo-document
      and extract a structured, answer-free retrieval
      intent.

Think of the pseudo-document as a noisy proposal:
- It may expose useful evidence wording, such as relation
      phrases, field names, role words, document-section
      words, or corpus-style expressions.
```

Method	LLM or Structured Output	Retrieval Text
Query	–	how many episodes are in Cinder Harbor season 4
QUERY2DOC	*Cinder Harbor season 4 has 12 episodes. The fourth season aired during the show's regular broadcast run, beginning with a premiere that introduced the marina fire storyline and ending with the finale "Low Tide."*	how many episodes are in Cinder Harbor season 4 [SEP] Cinder Harbor season 4 has 12 episodes. The fourth season aired during the show's regular broadcast run, beginning with a premiere that introduced the marina fire storyline and ending with the finale "Low Tide."

SAFEINTENT	{
	answer_type: number,
	anchors: [Cinder Harbor, season 4],
	relation_frame: {
	subject: Cinder Harbor season 4,
	relation: episode count,
	answer_slot: [NUMBER],
	query_constraints: [season 4]
	},
	bridge_cues: [episodes, fourth season, aired, broadcast run, premiere, finale],
	unsafe_spans: [12, marina fire storyline, Low Tide],
	confidence: high,
	retrieval_policy: answer_format_sketch
	}
		Sparse bridge view:
		how many episodes are in Cinder Harbor season 4 Cinder Harbor season 4 episodes fourth season aired broadcast run premiere finale
		Dense evidence sketches:
		1. Cinder Harbor season 4 has [NUMBER] episodes.
		2. The fourth season of Cinder Harbor consists of [NUMBER] episodes.
		3. The fourth season of Cinder Harbor aired [NUMBER] episodes during its broadcast run.
		4. Cinder Harbor's fourth season includes [NUMBER] episodes.

Table 7: Comparison between QUERY2DOC and SAFEINTENT on a plausible private-like renamed query. QUERY2DOC produces a fluent pseudo-document that fills in a concrete answer and additional unsupported context. SAFEINTENT distills bridge cues exposed by the pseudo-document, rejects unsafe spans, and renders answer-free sparse and dense retrieval views with the answer slot left blank.

```
- It may also contain unsupported concrete answers, answer
      aliases, or invented facts.

Return exactly one JSON object with this schema:
{
  "answer_type": "person|location|organization|date|number|
      title|event|entity",
  "query_intent": {
    "asked_relation": "short description of the relation
      asked by the query",
    "expected_answer_role": "short description of the
      missing answer role"
  },
  "anchors": [
    "important exact substring copied from the original
      query"
  ],
  "relation_frame": {
    "subject": "main query anchor or query-supported
      subject",
    "relation": "short query-supported relation phrase",
    "answer_slot": "[PERSON]|[LOCATION]|[ORGANIZATION]|[
      DATE]|[NUMBER]|[TITLE]|[EVENT]|[ENTITY]",
    "query_constraints": [
      "exact query substring that constrains the relation"
    ]
  },
  "bridge_cues": [
    {
      "text": "short generic answer-free evidence cue",
      "source_span": "exact substring copied from the
      pseudo-document"
    }
  ],
  "unsafe_spans": [
    {
      "text": "exact substring copied from the pseudo-
      document",
      "reason": "direct_answer|answer_alias|
      query_external_entity|unsupported_fact|
      specific_number_or_date|refusal_or_meta"
    }
  ],
  "confidence": "high|medium|low",
  "retrieval_policy": "answer_format_sketch|bridge_only|
      query_only"
}

How to identify anchors:
- Anchors are the important referring expressions in the
      query: the subject, work title, person, organization,
       location, event, product, code, or modifier that the
       answer must be about.
- Choose anchors by understanding the query, not by taking
      every noun.
- Each anchor must be an exact substring of the original
      query.
- Prefer complete names or titles over partial spans.
- Do not mark query anchors unsafe, even if they look
      synthetic or unfamiliar.

How to identify bridge cues:
- A bridge cue is not the answer.
- A bridge cue is wording from the pseudo-document that
      could help retrieve a corpus passage expressing the
      same relation.
- Bridge cues should be generic, answer-free relation or
      evidence expressions.
- Good bridge cues include relation wording, answer-field
      wording, evidence containers, role words, document-
      section words, and natural corpus phrasing.
- A bridge cue should remain useful even if the concrete
      answer in the pseudo-document is wrong.
- Each source_span must be an exact substring from the
      pseudo-document.
- text may be a short normalization of source_span, but it
      must not add new semantic content.
- Content appearing only in the pseudo-document is not
      trusted as fact. It may be used only as a generic,
      answer-free bridge cue; otherwise mark it unsafe or
      omit it.

How to identify unsafe spans:
- Unsafe spans are mainly discovered from the information
      gap between the query and the pseudo-document.
- If the pseudo-document supplies a concrete value that the
       query asks for, mark that value unsafe.
```

```
- If the pseudo-document introduces a query-external name,
      title, event, date, year, number, place, organization
      , or highly specific fact, mark it unsafe unless it
      is only a generic answer-free bridge cue.
- If a phrase could be either a useful bridge cue or an
      unsafe answer-specific fact, mark it unsafe or omit
      it.

How to build the relation frame:
- The subject must be copied from, or directly supported by
      , the original query.
- The relation must describe only the relation asked by the
       query.
- The relation must not contain concrete answer guesses,
      query-external entities, titles, dates, years,
      numbers, or factual assumptions from the pseudo-
      document.
- The answer_slot must match answer_type.

Confidence:
- Use "high" when the query relation is clear, anchors are
      clear, and at least one safe bridge cue is supported
      by the pseudo-document.
- Use "medium" when the relation is clear but bridge cues
      are sparse or partly generic.
- Use "low" when the pseudo-document is a refusal/meta
      response, the relation is ambiguous, or safe bridge
      cues cannot be separated from unsafe facts.

Retrieval policy:
- Use "answer_format_sketch" when the relation and anchors
      are clear enough to render answer-free evidence
      sketches.
- Use "bridge_only" when bridge cues are useful for lexical
       matching but not reliable enough for natural
      evidence sketches.
- Use "query_only" when the pseudo-document is a refusal/
      meta response or when no safe bridge cue remains.

Rules:
- Output JSON only. Do not output markdown, comments,
      explanations, or a passage.
- Do not answer the question.
- Do not use world knowledge to decide the answer.
- Preserve query anchors exactly.
- Do not introduce any concrete answer, entity, title, date
      , year, number, or factual claim unless it is already
       present in the original query.
- Bridge cues must not contain direct answers, answer
      aliases, query-external named entities, query-
      external titles, dates, years, exact numbers, or
      refusal-like text.
- Content appearing only in the pseudo-document may be used
       only as a generic, answer-free bridge cue; otherwise
       mark it unsafe or omit it.
- Keep at most 8 bridge cues. Prefer cues that are likely
      to appear in corpus passages.

Original query:
{query}

Query2Doc pseudo-document:
{query2doc}

JSON:
```

D.2 Without-Distill Relation-Term Prompt

This prompt is used for the SafeIntent without-distill ablation. It does not inspect the Query2Doc pseudo-document and instead generates generic answer-free relation terms directly from the original query.

Prompt 2: Without-Distill Relation-Term Intent

```
You are generating an answer-free retrieval intent for
      private RAG.

The query may contain plausible but renamed entities, such
      as synthetic show names, people, organizations,
      products, or places. Treat these names as real corpus
       anchors. Do not decide whether they are known
      entities. Do not answer the query from world
      knowledge.

You are given only the original query. Your task is to
      infer the query's retrieval intent from the query
      wording alone and generate generic, answer-free
      relation terms that may help retrieve passages
      expressing the requested relation.

Do not answer the query. Do not use world knowledge. Do not
       introduce any entity, fact, date, number, title, or
      answer that is not present in the original query.

Return exactly one JSON object with this schema:
{
  "answer_type": "person|location|organization|date|number|
      title|event|entity",
  "query_intent": {
    "asked_relation": "short description of the relation
      asked by the query",
    "expected_answer_role": "short description of the
      missing answer role"
  },
  "anchors": [
    "important exact substring copied from the original
      query"
  ],
  "relation_frame": {
    "subject": "main query anchor or query-supported
      subject",
    "relation": "short query-supported relation phrase",
    "answer_slot": "[PERSON]|[LOCATION]|[ORGANIZATION]|[
      DATE]|[NUMBER]|[TITLE]|[EVENT]|[ENTITY]",
    "query_constraints": [
      "exact query substring that constrains the relation"
    ]
  },
  "relation_terms": [
    "short generic answer-free relation or evidence phrase"
  ],
  "confidence": "high|medium|low",
  "retrieval_policy": "answer_format_sketch|bridge_only|
      query_only"
}

How to identify anchors:
- Anchors are the important referring expressions in the
      query: the subject, work title, person, organization,
       location, event, product, code, or modifier that the
       answer must be about.
- Choose anchors by understanding the query, not by taking
      every noun.
- Each anchor must be an exact substring of the original
      query.
- Prefer complete names or titles over partial spans.
- Preserve private codes and renamed entities exactly.

How to infer the relation from the query:
- Identify what information is missing from the query and
      what role the answer would fill.
- Use the wh-word or question form as a clue:
  - "who" often asks for a person, creator, performer,
      owner, member, or role;
  - "where" often asks for a location, origin, venue, or
      place;
  - "when" often asks for a date, release time, event time,
       or period;
  - "how many" often asks for a count or quantity;
  - "what is/are" often asks for an identity, definition,
      function, title, or entity.
- Use query constraints such as season numbers, dates,
      titles, roles, locations, or modifiers to specify the
       relation.
- The relation phrase must describe only the relation
      requested by the query.
- The relation phrase must not contain concrete answer
      guesses, query-external entities, titles, dates,
      years, numbers, or factual assumptions from world
      knowledge.

How to generate relation terms:
- Relation terms are not answers.
```

```
- Relation terms must be generic, answer-free evidence
      language inferred from the query relation.
- Good relation terms include relation words, answer-field
      wording, role words, evidence containers, document-
      section words, and natural corpus expressions.
- Relation terms should help retrieve passages that state
      the requested relation, even when the concrete answer
       is unknown.
- Relation terms must not contain concrete people,
      locations, organizations, titles, dates, years, exact
       numbers, or proper nouns not already in the query.
- Do not add any entity or fact that is not present in the
      query.
- Keep at most 6 relation terms.

Confidence:
- Use "high" when the query relation and anchors are clear
      and generic relation terms can be safely inferred
      from the query.
- Use "medium" when the relation is clear but relation
      terms are sparse or generic.
- Use "low" when the relation is ambiguous or safe relation
       terms cannot be inferred without world knowledge.

Retrieval policy:
- Use "answer_format_sketch" when the relation and anchors
      are clear enough to render answer-free evidence
      sketches.
- Use "bridge_only" when relation terms are useful for
      lexical matching but not reliable enough for natural
      evidence sketches.
- Use "query_only" when the relation is ambiguous or no
      safe relation term can be inferred.

Rules:
- Output JSON only. Do not output markdown, comments,
      explanations, or a passage.
- Do not answer the question.
- Do not use world knowledge to decide the answer.
- Preserve query anchors exactly.
- answer_type must describe the unknown answer being asked
      for.
- answer_slot must match answer_type.
- relation_frame.subject must be copied from, or directly
      supported by, the original query.
- relation_frame.query_constraints must be exact substrings
       from the original query.
- relation_terms must be generic evidence language only.
- relation_terms must not contain direct answers, answer
      aliases, query-external named entities, query-
      external titles, dates, years, exact numbers, or
      refusal-like text.
- If the relation cannot be inferred without adding outside
       knowledge, use confidence = "low", relation_terms =
      [], and retrieval_policy = "query_only".

Original query:
{query}

JSON:
```

D.3 Constrained Dense Renderer Prompt

This prompt renders a validated SAFEINTENT intent into dense answer-format sketches. Sparse retrieval is not generated by the LLM; the sparse bridge view is constructed deterministically by concatenating the original query, copied anchors, and sanitized bridge cues.

Prompt 3: SAFEINTENT Constrained Dense Renderer

```
You are a constrained dense retrieval-view renderer for
      private RAG.

You are not a question-answering system. You must not
      answer the query.
Your job is to turn a validated retrieval intent into short
      , answer-free,
corpus-style dense retrieval sketches.

Sparse retrieval is not generated by this prompt. The
      sparse bridge view is
constructed deterministically outside the LLM by
      concatenating:
original query + copied anchors + sanitized bridge cues.

You only generate dense evidence sketches.

You are given the following inputs:

1. Original query
   The user's original information need. Use it only to
      preserve the query
   meaning, anchors, and constraints. Do not answer it.

2. Answer type
   The type of the missing answer, such as person, location
      , date, number,
   title, event, or entity.

3. Anchors
   Exact query substrings that identify the retrieval
      target. These must be
   copied exactly into each sketch when relevant. Anchors
      may be renamed or
   unfamiliar, but they are real corpus anchors, not
      placeholders.

4. Relation frame
   A structured description of what relation the query asks
       for:
   - subject: the main query-supported entity or target;
   - relation: the relation that should be expressed in the
       sketch;
   - answer_slot: the typed placeholder for the missing
      answer, such as
     [PERSON], [LOCATION], [DATE], [NUMBER], [TITLE], [
      EVENT], or [ENTITY];
   - query_constraints: query substrings that constrain the
       relation, such as
     season numbers, dates, roles, locations, or modifiers.

5. Sanitized bridge cues
   Safe relation-level wording distilled from the pseudo-
      document. These cues
   are the main source of answer-format alignment. They may
       include relation
   words, field names, evidence containers, role words,
      document-section words,
   or corpus-style phrases. They are not facts and must not
       be treated as
   answers.

6. Unsafe spans
   Strings that must never appear in the output. These may
      include generated
   answers, unsupported facts, query-external entities,
      titles, numbers, dates,
   or refusal/meta text.

How to render dense evidence sketches:

Each sketch should look like a short sentence that could
      appear in a relevant
corpus passage. It should express the relation in the
      relation frame while
leaving the answer blank with the answer_slot.

A good sketch should:
- preserve the main query anchors exactly;
- preserve the relation described in the relation frame;
- preserve relevant query constraints;
- place the answer_slot where the missing answer would
      appear;
- use sanitized bridge cues to make the sketch resemble
      corpus evidence;
- avoid adding any concrete answer or query-external fact.

Use bridge cues actively:
- Use as many sanitized bridge cues as can be included
      naturally and safely.
- Prefer bridge cues that express the relation, answer
      field, evidence location,
  or corpus-style wording.
```

```
- It is acceptable to combine multiple bridge cues in one
      sketch if the sentence
  remains natural and answer-free.
- Omit a bridge cue only if it is redundant, unnatural in
      the sentence, unsafe,
  or would make the sketch unsupported.
- Do not invent new bridge cues that are not supported by
      the input.

Return exactly one JSON object with this schema:
{
  "dense_evidence_sketches": [
    "one short answer-free corpus-style evidence sentence"
  ],
  "used_bridge_cues": [
    "bridge cue copied from the input"
  ],
  "omitted_bridge_cues": [
    "bridge cue copied from the input"
  ]
}

Output requirements:
- Output JSON only. Do not output markdown, comments,
      explanations, or caveats.
- Generate 2 to 4 dense evidence sketches when possible.
- Each sketch should be a complete natural sentence.
- Each sketch must preserve the query relation.
- Each sketch must include the answer_slot from the
      relation frame.
- Each sketch must preserve important query anchors exactly
      .
- Each sketch should preserve relevant query constraints.
- Each sketch should use at least one bridge cue when a
      safe bridge cue is
  available.
- Across all sketches, use as many non-redundant safe
      bridge cues as possible.
- If a bridge cue is used, copy it into used_bridge_cues.
- If a bridge cue is not used, copy it into
      omitted_bridge_cues.
- If no sketch can be rendered safely, return an empty
      dense_evidence_sketches
  list.

Safety rules:
- Do not answer the query.
- Do not output a concrete answer.
- Do not introduce any person, location, organization,
      title, event, date,
  year, number, or factual claim unless it already appears
      in the original query.
- Do not include any unsafe span or part of an unsafe span.
- Do not say that an entity is private, fictional, renamed,
       unknown, or a
  placeholder.
- Do not use world knowledge.

Original query:
{query}

Answer type:
{answer_type}

Anchors:
{anchors_json}

Relation frame:
{relation_frame_json}

Sanitized bridge cues:
{bridge_cues_json}

Unsafe spans:
{unsafe_spans_json}

JSON:
```

D.4 Without-Distill Dense Renderer Prompt

This prompt is used for the SAFEINTENT without-distill ablation. It has the same rendering interface as the dense renderer above, but the input cues are generic relation terms inferred from the original query rather than bridge cues distilled from a QUERY2DOC pseudo-document.

Prompt 4: Without-Distill Constrained Dense Renderer

```
You are a constrained dense retrieval-view renderer for
      private RAG.

You are not a question-answering system. You must not
      answer the query.
Your job is to turn a validated query-only retrieval intent
       into short,
answer-free, corpus-style dense retrieval sketches.

This is the without-distill ablation. You are not given a
      Query2Doc
pseudo-document, and you must not assume any evidence
      wording from one.
The relation terms are inferred only from the original
      query. They are generic
relation cues, not corpus-grounded bridge cues and not
      facts.

Sparse retrieval is not generated by this prompt. The
      sparse bridge view is
constructed deterministically outside the LLM by
      concatenating:

original query + copied anchors + safe relation terms.

You only generate dense evidence sketches.

You are given the following inputs:

1. Original query
   The user's original information need. Use it only to
      preserve the query
   meaning, anchors, and constraints. Do not answer it.

2. Answer type
   The type of the missing answer, such as person, location
      , date, number,
   title, event, or entity.

3. Anchors
   Exact query substrings that identify the retrieval
      target. These must be
   copied exactly into each sketch when relevant. Anchors
      may be renamed or
   unfamiliar, but they are real corpus anchors, not
      placeholders.

4. Relation frame
   A structured description of what relation the query asks
       for:
   - subject: the main query-supported entity or target;
   - relation: the relation that should be expressed in the
       sketch;
   - answer_slot: the typed placeholder for the missing
      answer, such as
     [PERSON], [LOCATION], [DATE], [NUMBER], [TITLE], [
      EVENT], or [ENTITY];
   - query_constraints: query substrings that constrain the
       relation, such as
     season numbers, dates, roles, locations, or modifiers.

5. Safe relation terms
   Generic answer-free relation wording inferred from the
      original query.
   These terms may include relation words, role words,
      answer-field wording,
   or common evidence expressions. They are not facts, not
      answers, and not
   pseudo-document evidence.

How to render dense evidence sketches:

Each sketch should look like a short sentence that could
      appear in a relevant
corpus passage. It should express the relation in the
      relation frame while
leaving the answer blank with the answer_slot.
```

```
A good sketch should:
- preserve the main query anchors exactly;
- preserve the relation described in the relation frame;
- preserve relevant query constraints;
- place the answer_slot where the missing answer would
      appear;
- use safe relation terms to make the relation explicit;
- avoid adding any concrete answer or query-external fact.

Use relation terms actively:
- Use as many safe relation terms as can be included
      naturally and safely.
- Prefer relation terms that clarify the relation, answer
      field, role, count,
  location, time, definition, membership, ownership, or
      other queried relation.
- It is acceptable to combine multiple relation terms in
      one sketch if the
  sentence remains natural and answer-free.
- Omit a relation term only if it is redundant, unnatural
      in the sentence,
  unsafe, or would require adding information not present
      in the query.
- Do not invent new entities, facts, or corpus-specific
      details beyond the input.
- Do not treat relation terms as evidence from a pseudo-
      document; they are only
  query-inferred wording cues.

Return exactly one JSON object with this schema:
{
  "dense_evidence_sketches": [
    "one short answer-free corpus-style evidence sentence"
  ],
  "used_relation_terms": [
    "relation term copied from the input"
  ],
  "omitted_relation_terms": [
    "relation term copied from the input"
  ]
}

Output requirements:
- Output JSON only. Do not output markdown, comments,
      explanations, or caveats.
- Generate 2 to 4 dense evidence sketches when possible.
- Each sketch should be a complete natural sentence.
- Each sketch must preserve the query relation.
- Each sketch must include the answer_slot from the
      relation frame.
- Each sketch must preserve important query anchors exactly
      .
- Each sketch should preserve relevant query constraints.
- Each sketch should use at least one relation term when a
      safe relation term is
  available.
- Across all sketches, use as many non-redundant safe
      relation terms as possible.
- If a relation term is used, copy it into
      used_relation_terms.
- If a relation term is not used, copy it into
      omitted_relation_terms.
- If no sketch can be rendered safely, return an empty
      dense_evidence_sketches
  list.

Safety rules:
- Do not answer the query.
- Do not output a concrete answer.
- Do not introduce any person, location, organization,
      title, event, date,
  year, number, or factual claim unless it already appears
      in the original query.
- Do not say that an entity is private, fictional, renamed,
       unknown, or a
  placeholder.
- Do not use world knowledge.
- Do not add corpus-specific details, section names, titles
      , aliases, or facts
  that are not supported by the original query or the
      validated relation frame.

Original query:
{query}

Answer type:
{answer_type}

Anchors:
{anchors_json}

Relation frame:
{relation_frame_json}

Safe relation terms:
{relation_terms_json}

JSON:
```

E Artifact Licenses and Distribution

We use public benchmarks, pretrained embedding models, and hosted language-model APIs only under their respective licenses, terms of use, and access conditions. We do not redistribute third-party corpora, model weights, API outputs, generated pseudo-documents, or cached artifacts beyond aggregate results. All external artifacts are used solely for research evaluation.

F Intended Use of Artifacts

Artifacts are used only to study retrieval behavior in public and synthetically renamed private-like settings. Benchmarks support evaluation, embedding models support dense retrieval, and LLM APIs support query-expansion and SAFEINTENT artifact generation. Synthetic renamings are artificial and do not represent real private records. Prompts, generated intents, renamed examples, and evaluation results are intended only for research on private-like RAG and leakage-free retrieval reformatting.