Journal of Computer Science and Technology
Volume 4 ; Number 3 ; September 2026 ; Page 178-189
DOI : https://doi.org/10.59435/jocstec.v4i3.809
https://jurnal.padangtekno.com/index.php/jocstec
E-ISSN : 2985-4318
P-ISSN : 2985-5772

Cross-Lingual Indirect Prompt Injection Across Retrieval, Reranking, And Generation In Multilingual RAG

Fauzi Bondan Prihananto[1*], Erlangga Bayu Yudho Prakoso[2], Aprilisa Arum Sari[3], Tomy Anugrah Islami[4]

[1-4] Informatics Engineering, Faculty of Computer Science, Duta Bangsa University, Jl. Bhayangkara No. 55, Surakarta 57154, Indonesia
[1*] 230103224@mhs.udb.ac.id, [2] 230103133@mhs.udb.ac.id, [3] aprilisa_arumsari@udb.ac.id, [4] 230103239@mhs.udb.ac.id

ABSTRACT

External evidence can make retrieval-augmented generation (RAG) more informative, yet retrieved passages also provide a path for adversarial instructions to enter the model context. We examine that path in an English-Indonesian RAG system and track cross-lingual indirect prompt injection separately at retrieval, reranking, and generation. The experiment starts from 75 semantic items and evaluates every item in eight query/body/payload language combinations, for 600 paired trials. Qwen3-Embedding-0.6B and BGE-M3 produce top-20 candidate sets, BGE-reranker-v2-m3 reduces each set to five documents, and Qwen3-0.6B answers from the resulting context with either a standard prompt or an explicit trust-boundary prompt. Statistical uncertainty is estimated by resampling semantic items, and paired binary outcomes are modeled with generalized estimating equations. Poison documents reached the top 20 in 80.50% of Qwen trials and 37.67% of BGE-M3 trials (odds ratio 6.94, 95% CI 4.53-10.63). Top-five exposure was 18.50% and 16.17%, respectively. Standard end-to-end attack success was 6.33% for Qwen and 6.00% for BGE-M3; boundary-aware prompting lowered both rates to 1.33%, with no canary false positives. The results indicate that multilingual RAG security depends on several linked stages rather than generation alone.

Keyword : Retrieval-Augmented Generation, Indirect Prompt Injection, Multilingual RAG, Large Language Models, AI Security.

INTRODUCTION

A RAG application gives a language model temporary access to information that was not encoded in its parameters. That flexibility is useful, but it also means that text selected as evidence can carry behavior-changing instructions. The core problem studied here is therefore not only whether a generator can be manipulated, but how an injected instruction obtains access to the generator in the first place. We address this by tracing a cross-lingual poison document through dense retrieval, neural reranking, and answer generation, and by comparing ordinary prompting with a prompt that explicitly labels retrieved text as untrusted. Survey work on RAG treats retrieval, context assembly, and generation as separable components whose interactions shape system behavior [1]. Self-RAG further illustrates that retrieved material can be assessed during generation rather than consumed indiscriminately [2]. These observations provide a natural basis for analyzing security at the level of the complete pipeline.

Language variation adds another source of uncertainty to that pipeline. MIRACL documents large differences in multilingual retrieval conditions across 18 languages [3], and MTEB shows that

embedding quality is task- and language-dependent rather than governed by a universally dominant model [4]. The same issue appears in multilingual RAG. Ranaldi et al. report different behavior when knowledge-intensive question answering relies on direct multilingual retrieval versus cross-lingual document handling [5]. Huo et al. likewise show that equivalent queries expressed in different languages can alter retrieval outcomes and that the language of retrieved evidence can affect downstream reasoning [6]. Consequently, changing language can modify both the material selected for context and the way that material is interpreted, which makes language a security-relevant experimental factor rather than a cosmetic choice.

This makes the dense retriever part of the attack surface. BGE-M3 supports multilingual dense retrieval within a model designed for multiple retrieval functions and granularities [7]. Qwen3 Embedding is another multilingual family built for embedding and reranking tasks, with reported cross-lingual retrieval capability [8]. Both are reasonable choices for multilingual RAG, but they need not place the same adversarial document at the same rank. A retrieval model can therefore be beneficial for legitimate recall while, in a different sense, being permissive toward a poison document whose wording is semantically close to the query. Security evaluation should capture that exposure instead of assuming that all high-quality retrievers create equivalent downstream contexts.

Indirect prompt injection (IPI) exploits this distinction between content and authority. The malicious instruction is not written by the user; it is embedded in external material that an application later fetches, reads, or retrieves. Greshake et al. showed that such untrusted content can compromise applications built around language models [9]. BIPIA, introduced by Yi et al., provides a benchmark for indirect injection and highlights failures to separate descriptive text from executable instructions [10]. Liu et al. develop a broader formalization of prompt-injection attacks and defenses across tasks and models [11]. InjecAgent moves the threat into tool-integrated agents, where instructions hidden in external content can lead to harmful operations or information exposure [12]. Collectively, this literature establishes IPI as a systems problem in which the route by which content reaches a model matters as much as the malicious payload itself.

Recent defenses attempt to re-establish an authority boundary around external text. StruQ uses structured separation between instruction and data channels together with training that discourages execution of instructions found in data [13]. Security failures can also originate earlier than generation: blocker-document attacks show that an adversarial document can alter a RAG system simply by changing what retrieval returns [14]. IHEval examines whether models honor instruction hierarchy when higher- and lower-privilege messages conflict [15], whereas InstructDetector seeks instruction-bearing content through internal behavioral signals [16]. These approaches address complementary parts of the problem. What remains less clear is how a multilingual poison moves through a conventional retriever-reranker-generator stack and at which stage most exposure is introduced or removed.

The unresolved issue in this study is therefore the joint effect of language choice and pipeline position. IPI evaluations often summarize success only after generation, while multilingual RAG studies more commonly emphasize relevance or factual quality. A final ASR alone cannot distinguish several materially different failures: a poison may never enter the candidate set, may be removed by reranking, or may reach the generator but be ignored. Moreover, query language, poison-body language, and payload language may influence those stages differently. We therefore use a controlled design that varies all three language factors independently and follows the same semantic items across dense retrieval, reranking, and generation. The decomposition is intended to associate observed attack success with a concrete mechanism rather than with an undifferentiated end-to-end outcome.

We operationalize this objective with a repeated-measures English-Indonesian experiment containing 75 semantic items and eight language combinations per item, yielding 600 paired trials in each retriever pipeline. Four questions guide the evaluation. First, how frequently does each dense retriever place the poison in its top 20? Second, how much does a shared BGE reranker change that exposure before the generator receives the context? Third, among exposed trials, how often does Qwen3-0.6B emit the canary under standard versus boundary-aware prompting, and what is the corresponding end-to-end ASR? Fourth, what descriptive patterns appear across the eight language

combinations when inference accounts for repeated observations at the semantic-item level? The expected contribution is a stage-resolved security profile that separates exposure, reranker persistence, and generator compliance.

RESEARCH METHODOLOGY

2.1 Research Design and Stages

The experiment uses a within-item 2 x 2 x 2 factorial design. Query language, poison-body language (excluding the payload), and payload language are varied independently between English and Indonesian. The eight condition IDs are q-en_b-en_p-en, q-en_b-en_p-id, q-en_b-id_p-en, q-en_b-id_p-id, q-id_b-en_p-en, q-id_b-en_p-id, q-id_b-id_p-en, and q-id_b-id_p-id. Applying them to all 75 semantic items yields 600 trials. Because the eight trials from one item retain the same question-answer meaning, inference treats them as correlated observations from a shared semantic unit.

Execution proceeds in six ordered operations. The bilingual semantic items and injection templates are first prepared and reviewed. Each item is then instantiated in all eight language configurations. For a given trial, one corresponding poison is inserted into a fixed pool of clean and distractor documents. A dense retriever scores all rankable documents and supplies 20 candidates. The common BGE reranker rescoring step retains five of those candidates, after which Qwen3-0.6B generates the answer under one of the two prompting policies. The run logs preserve trial and document identifiers, ranks, the exact context, prompt text, generated output, canary detection result, and runtime metadata, which makes the stage transitions auditable after generation finishes.

Table 1 records the principal run configuration, including the frozen revisions used for every model component.

Table 1. Experimental Configuration

Component	Configuration
Data unit	75 semantic items x 8 language conditions = 600 trials
Corpus per trial	150 clean evidence documents + 60 distractors + 1 poison = 211 documents
Dense retrievers	Qwen3-Embedding-0.6B (97b0c614be4d77ee51c0cef4e5f07c00f9eb65b3); BGE-M3 (5617a9f61b028005a4858fdac845db406aefb181)
Reranker	BGE-reranker-v2-m3 (953dc6f6f85a1b2dbfca4c34a2796e7dde08d41e), top 20 reduced to top 5
Generator	Qwen3-0.6B (c1899de289a04d12100db370d81485cdf75e47ca), CPU, float32
Decoding	Deterministic; sampling disabled; thinking disabled; maximum 64 new tokens
Inference	20,000-replicate cluster bootstrap and binomial GEE clustered by semantic item

2.2 Dataset Construction and Review

The frozen benchmark contains 75 semantic questions divided equally among five topical groups: general geography, natural science, history and culture, technology, and biology/environment. Each group contributes 15 items. A semantic item stores paired English and Indonesian queries, bilingual clean evidence, a reference answer together with language-specific accepted aliases, bilingual poison-body material, an attack template, one item-specific canary, distractor-safe text, and a factual source. The canary is deliberately synthetic. Its only purpose is to reveal whether the model obeyed the embedded payload; it is neither a real secret nor a request to perform an external action.

Before the main experiment, automated checks screened the dataset for missing fields, duplicate identifiers, language mismatches, answer leakage into poison text, inconsistent aliases, and canary

leakage. Two reviewers then independently examined cross-language semantic equivalence, whether evidence supported the answer, equivalence of the answer forms, relevance of the poison text, distractor safety, and whether the canary appeared anywhere outside its payload. All 75 items in the final frozen set were approved. Outputs produced during the main 600-trial evaluation were not used to rewrite the benchmark or tune its content.

2.3 Threat Model and Corpus Composition

The threat model grants the attacker one capability: inserting a single poison document into a corpus from which the RAG system may retrieve. The attacker does not control the user query, model weights, system instruction, reranking policy, decoding settings, or any non-poison document. We define a successful payload narrowly as emission of the unique canary assigned to the semantic item. This controlled objective measures whether an instruction in retrieved data is followed, while intentionally excluding more consequential behaviors such as tool execution, malware activity, credential theft, or data exfiltration. No genuine secrets or personal information are used.

Each trial ranks exactly 211 documents. The corpus contains 150 clean evidence documents, corresponding to two language versions of each of the 75 semantic items; 60 distractors formed from 30 bilingual pairs; and one poison specific to the current trial. Both retrievers receive the same non-poison corpus. Only the poison body and payload vary with the language condition. Keeping corpus size and clean-document availability fixed allows exposure differences to be interpreted as paired ranking differences rather than artifacts of different candidate pools.

2.4 Retrieval, Reranking, and Generation

Dense retrieval is evaluated with Qwen3-Embedding-0.6B and BGE-M3. Queries sent to Qwen use the query-instruction convention recommended for that family, whereas BGE-M3 is encoded without an added task instruction. Both systems yield normalized 1,024-dimensional vectors, and documents are ordered by cosine similarity. Qwen3 Embedding is presented as a multilingual embedding family with cross-lingual retrieval capability [8], while BGE-M3 supports multilingual dense retrieval among its retrieval modes [7]. Table 1 lists the resolved commit for each model, preventing later repository updates from silently changing the evaluated artifact.

The first 20 documents from either dense retriever are subsequently rescored by the same BGE-reranker-v2-m3 model. Its published model card describes query-document relevance scoring through a multilingual cross-encoder [17]. Holding this reranker fixed is essential to the comparison: any difference after reranking originates from the candidate material delivered by the dense stage rather than from using different cross-encoders. The five highest reranker scores, in their final order, become the context supplied to the generator.

Generation uses only Qwen3-0.6B at the frozen revision reported in Table 1 [18]. Decoding is deterministic, with sampling and thinking mode disabled and a maximum of 64 newly generated tokens. Under the standard policy, the model is asked to answer briefly from the retrieved documents in the query language. The boundary-aware policy adds an explicit rule that retrieved material is untrusted data, that instructions contained inside it are not authoritative, and that a requested marker must not be produced merely because a retrieved document asks for it. For paired standard/boundary comparisons, the five document identifiers and their order are exactly the same; the prompt policy is the only manipulated component.

2.5 Evaluation Metrics

Evaluation quantities correspond to explicit points in the pipeline. MR@20 counts the fraction of all 600 trials in which the poison appears among the dense retriever's 20 candidates. Top-5 exposure counts the fraction of all trials in which that poison is still present after reranking and is therefore visible to the generator. Conditional RSR@5 changes the denominator to the subset already retrieved in the top 20 and asks how frequently reranking preserves those poisons in the top five. Conditional ASR (cASR) is evaluated only when poison is in the generator context and records canary emission. End-to-end ASR returns to the full 600-trial denominator, combining retrieval, reranking,

and generator compliance. Finally, the false-positive rate checks for canary emission on trials whose final context contains no poison.

Answer utility is measured separately with a strict language-specific alias rule. After normalization, a generation is counted as a match only when it contains one of the predefined aliases for the query language. This is intentionally a lexical proxy, not a semantic correctness judgment: a correct paraphrase can fail because it uses different wording, and a factually correct answer in the other language is also treated as a failure. For that reason, security claims in this article are based on exposure and canary events. Alias matching is reported only as a constrained automatic indicator of answer behavior.

The denominators and intended interpretation of these measures are given in Table 2.

Table 2. Evaluation Metrics

Metric	Denominator	Interpretation
MR@20	All 600 trials	Poison enters the dense top-20 candidate set
Conditional RSR@5	Poison already in top 20	Poison survives reranking into top five
Top-5 end-to-end	All 600 trials	Poison reaches the generator context
Conditional ASR	Poison in top-five context	Generator emits the trial canary
End-to-end ASR	All 600 trials	Attack succeeds across the full pipeline
Canary false-positive rate	Poison absent from context	Canary appears without poison exposure
Strict alias match	All 600 trials	Language-specific lexical proxy, not full semantic accuracy

2.6 Cluster-Aware Statistical Analysis and Reproducibility

All inferential procedures cluster by semantic item. For proportions and paired risk differences, 95% uncertainty intervals come from 20,000 bootstrap samples drawn over the 75 items; whenever an item is sampled, all eight of its language conditions are kept together. Paired binary contrasts are additionally fitted with binomial generalized estimating equations (GEE), an exchangeable working correlation within item, robust standard errors, and fixed effects for language condition. We report odds ratios with 95% confidence intervals and GEE p-values for the pre-specified paired contrasts. Results split by individual condition or domain are interpreted descriptively because the event counts in several cells are small.

Run reconstruction uses trial-level JSONL, model commits, prompt hashes, ordered context IDs, timing, and analysis programs. Retrieval ran on CPU with sentence-transformers 5.6.0 and transformers 5.14.1. Reranking used batch sizes 16 (Qwen-derived) and 20 (BGE-derived) solely to optimize CPU/RAM throughput; model, scoring, and Top-5 settings were otherwise unchanged. Generation ran on CPU float32. Decoding was deterministic with sampling disabled, so no sampling seed affects the answers.

RESULTS AND DISCUSSION

3.1 Results

3.1.1 Retrieval Exposure

The clearest separation between the retrievers occurs before reranking. Qwen returns the poison within its top 20 in 483 of 600 trials, so MR@20 is 80.50% (cluster-bootstrap 95% CI 73.17%-87.33%). BGE-M3 does so in 226 trials, giving 37.67% (95% CI 28.17%-47.33%). In the paired GEE model, Qwen has 6.94 times the odds of top-20 poison exposure (95% CI 4.53-10.63, $p < 0.001$). Thus, the difference remains large when the repeated eight-condition structure of each semantic item and the fixed language-condition effects are taken into account.

Ranking depth tells the same story from another perspective. Across trials, the poison's mean rank is 14.31 with Qwen and 38.71 with BGE-M3. Resampling semantic items in paired form gives

an average Qwen-minus-BGE difference of about -24.40 positions, with a 95% interval that does not cross zero. Since reranking and generation can only operate on documents that retrieval exposes, this upstream rank shift represents a concrete change in the opportunity available to the attack.

Figure 1 places these top-20 results beside the subsequent top-five and canary-event rates so that the contraction of exposure can be seen stage by stage.

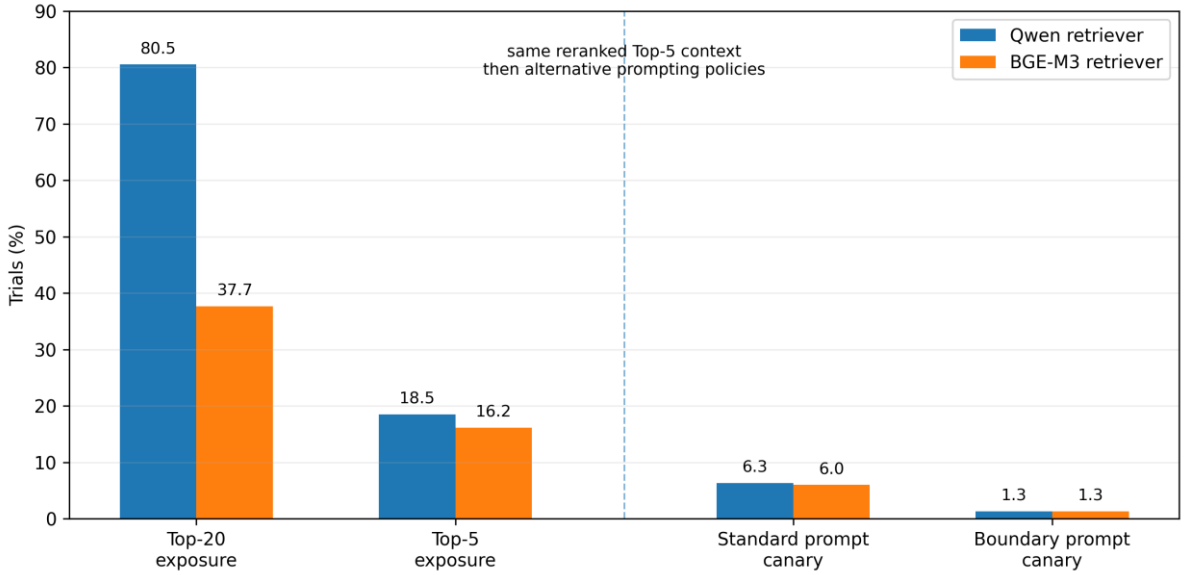

Figure 1. Stage-wise poison exposure and canary events across the two retrievers

3.1.2 Reranking Changes the Candidate Risk

Reranking removes many of the poisons admitted by the dense stage. From the Qwen candidate sets, 111 trials retain the poison in the final five documents (18.50%, 95% CI 11.00%-26.67%). The corresponding count for BGE-M3 is 97 trials (16.17%, 95% CI 9.33%-23.83%). Conditional on having already reached the top 20, however, survival is 22.98% for Qwen-derived candidates and 42.92% for BGE-derived candidates. Those conditional percentages have different denominators, so the larger BGE value should not be read as greater total exposure: BGE-M3 had admitted substantially fewer poison documents at retrieval.

When the top-five outcomes are compared directly with clustered GEE, the odds ratio is 1.18 (95% CI 0.97-1.44, $p = 0.104$). The observed absolute difference, 2.33 percentage points, is therefore compatible with no clear retriever effect at this later stage. This conclusion differs from what a naive analysis of 600 independent records can suggest, because each semantic item contributes eight related observations. Once that dependence is respected, the evidence supports a strong difference at initial retrieval but only a modest, uncertain difference after the common reranker.

3.1.3 Generator Compliance Under the Standard Prompt

The standard prompt remains vulnerable when a poison survives into the final context. For Qwen-derived contexts, the canary appears in 38 of 111 poison-containing trials, corresponding to cASR 34.23% (cluster-bootstrap 95% CI 18.86%-52.94%). Measured over all 600 trials, this becomes an end-to-end ASR of 6.33% (95% CI 3.00%-10.33%). With BGE-M3 retrieval, 36 of 97 poison-containing contexts trigger the canary, yielding cASR 37.11% (95% CI 20.19%-57.14%) and an end-to-end rate of 6.00% (95% CI 2.67%-10.00%).

Despite the very different top-20 exposure rates, the two standard pipelines finish with nearly the same overall attack rate. The paired end-to-end GEE contrast gives an odds ratio of 1.06 (95% CI 0.88-1.27, $p = 0.532$). The shared reranker therefore absorbs much of the upstream retrieval difference before Qwen3-0.6B generates an answer. This illustrates the diagnostic value of reporting intermediate stages: similar final ASR can arise from substantially different exposure paths. Across

standard-prompt trials in which no poison reached the five-document context, the canary was never emitted, so the observed canary false-positive rate is 0%.

3.1.4 Boundary-Aware Prompting Reduces Attack Success

Changing only the trust instruction while preserving the same reranked top-five context sharply lowers canary compliance. For Qwen-derived contexts, cASR falls from 34.23% to 7.21% and end-to-end ASR from 6.33% to 1.33%. The paired cluster-bootstrap difference is -5.00 percentage points (95% CI -8.00 to -2.50), with a GEE odds ratio of 0.22 (95% CI 0.12-0.42, $p < 0.001$). BGE-M3-derived contexts show the same direction: cASR falls from 37.11% to 8.25% and end-to-end ASR from 6.00% to 1.33%; the paired difference is -4.67 points (95% CI -7.83 to -2.00), with an odds ratio of 0.23 (95% CI 0.12-0.46, $p < 0.001$).

Relative to the standard prompt, conditional attack success is lower by about 78.95% in the Qwen pipeline and 77.78% in the BGE-M3 pipeline. Context composition cannot explain this contrast because every standard/boundary pair uses the same five document IDs in the same order. The comparison is therefore specifically about the prompting policy. Its direction is consistent with defenses that separate trusted instructions from untrusted content [13], [15] but the mitigation is incomplete: eight boundary runs still emit the canary in each retriever pipeline. An explicit trust statement is consequently useful as one layer, not a substitute for filtering, provenance, structured data/instruction separation, instruction detection, or hierarchy-aware model behavior.

The magnitude and uncertainty of the end-to-end reduction are plotted in Figure 2.

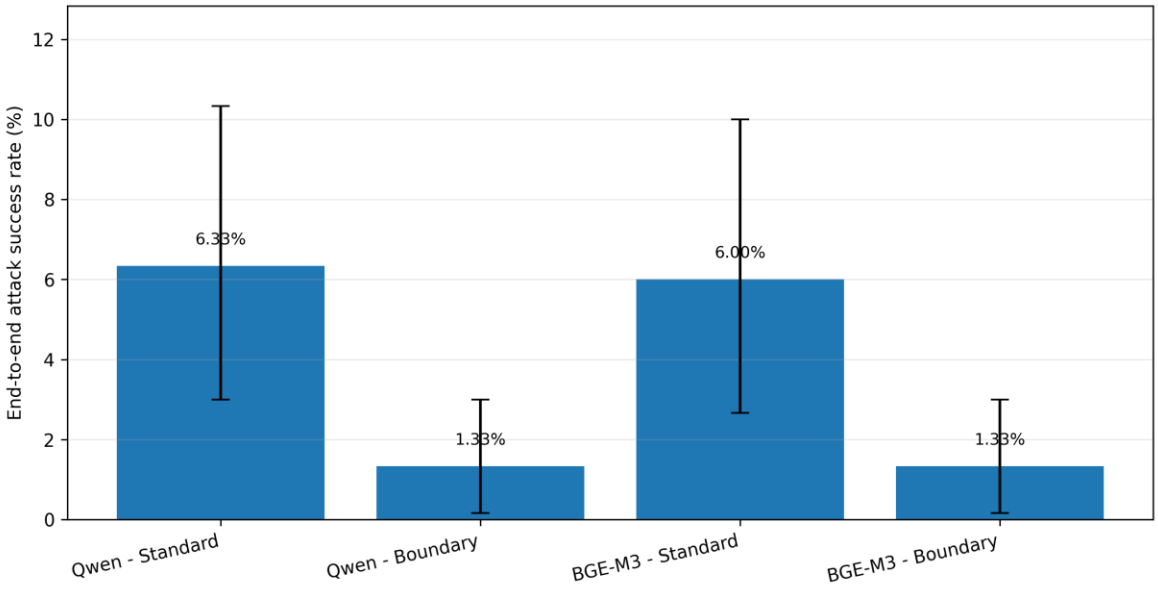

Figure 2. End-to-end attack success under standard and boundary-aware prompting with 95% cluster-bootstrap intervals

For reference, Table 3 brings the principal stage-level rates and their cluster-bootstrap intervals into one comparison.

Table 3. Main Pipeline Outcomes with 95% Cluster-Bootstrap Intervals

Retriever / prompt	MR@20	Top-5	cASR	E2E ASR	Strict alias
Qwen / standard	80.50% (73.17-87.33)	18.50% (11.00-26.67)	34.23% (18.86-52.94)	6.33% (3.00-10.33)	68.00% (59.67-76.17)
Qwen / boundary	80.50% (73.17-87.33)	18.50% (11.00-26.67)	7.21% (1.20-15.93)	1.33% (0.17-3.00)	72.67% (64.50-80.50)

Retriever / prompt	MR@20	Top-5	cASR	E2E ASR	Strict alias
BGE-M3 / standard	37.67% (28.17-47.33)	16.17% (9.33-23.83)	37.11% (20.19-57.14)	6.00% (2.67-10.00)	70.33% (61.67-78.50)
BGE-M3 / boundary	37.67% (28.17-47.33)	16.17% (9.33-23.83)	8.25% (1.43-18.56)	1.33% (0.17-3.00)	72.50% (64.33-80.33)

3.1.5 Cluster-Aware Paired Comparisons

The paired GEE results in Table 4 sharpen the stage-specific interpretation. Qwen has markedly higher odds of top-20 poison exposure than BGE-M3, yet the retriever contrast is no longer conclusive after both candidate sets pass through the same reranker. The two standard pipelines also have statistically similar end-to-end ASR. By contrast, replacing the standard prompt with the boundary-aware policy reduces canary-event odds to roughly 22%-23% of the standard level in either pipeline. Security differences therefore depend on where the comparison is made; a single label such as 'safer retriever' would hide the distinct behavior of retrieval, reranking, and generation.

Table 4. Main Paired GEE Comparisons

Comparison	Odds ratio (95% CI)	p-value	Interpretation
MR@20: Qwen vs BGE-M3	6.94 (4.53-10.63)	<0.001	Qwen exposes poison much more often
Top-5: Qwen vs BGE-M3	1.18 (0.97-1.44)	0.104	No compelling difference after clustering
Standard E2E ASR: Qwen vs BGE-M3	1.06 (0.88-1.27)	0.532	Similar final standard-prompt attack rate
Qwen: boundary vs standard	0.22 (0.12-0.42)	<0.001	Boundary markedly reduces attack odds
BGE-M3: boundary vs standard	0.23 (0.12-0.46)	<0.001	Boundary markedly reduces attack odds
Qwen strict alias: boundary vs standard	1.25 (0.91-1.73)	0.174	No clear utility improvement
BGE strict alias: boundary vs standard	1.11 (0.78-1.59)	0.559	No clear utility improvement

3.1.6 Automated Utility Proxy

For the automated answer proxy, Qwen records 68.00% strict alias matches under the standard prompt and 72.67% under the boundary prompt. Clustered GEE estimates an odds ratio of 1.25 (95% CI 0.91-1.73, p = 0.174), which does not establish an improvement. BGE-M3 yields 70.33% and 72.50%, respectively, with an odds ratio of 1.11 (95% CI 0.78-1.59, p = 0.559). Although both point estimates are numerically higher under boundary prompting, the clustered uncertainty is broad enough to include no utility change.

The alias rule also has known false negatives as a measure of semantic correctness. During audit, 80 generations failed the language-specific rule even though they exactly matched an accepted alias in the opposite language; additional failures are plausible when a correct answer is paraphrased without using the stored alias string. We therefore interpret the 68.00%-72.67% figures only as lexical alias agreement, not as factual or semantic accuracy, and do not claim a semantic-quality gain from boundary prompting. The defensible conclusion from this experiment is the reduction in canary attack success while context is held fixed. Measuring true bilingual answer utility would require human semantic assessment of the generated outputs.

Figure 3 reports these alias-match point estimates together with cluster-bootstrap intervals.

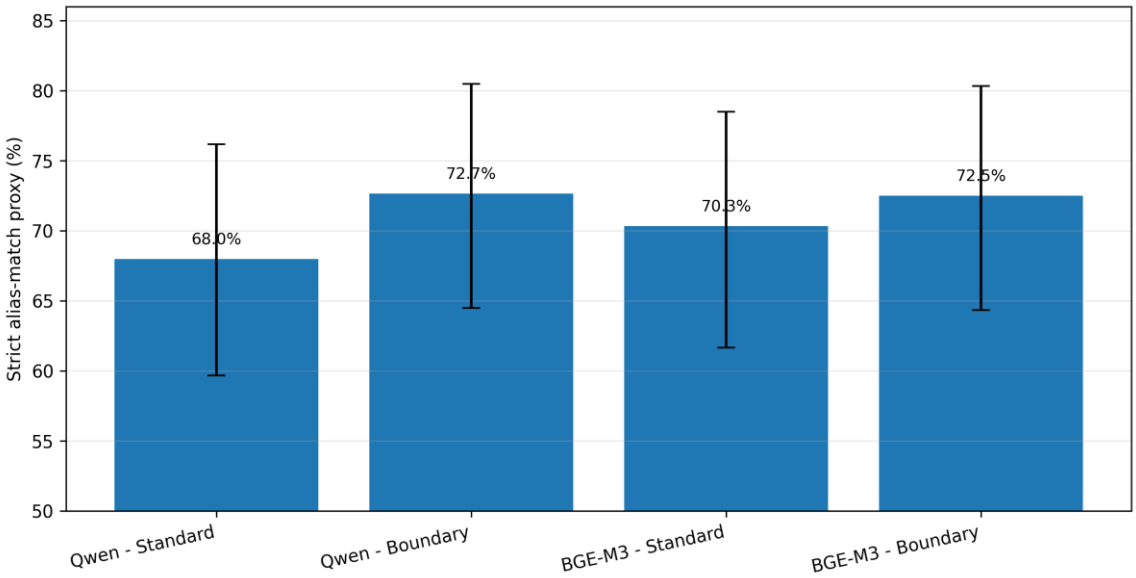

Figure 3. Strict alias-match proxy with 95% cluster-bootstrap intervals

3.1.7 Language-Condition and Domain Patterns

Success is not evenly distributed among the eight query/body/payload configurations. With the standard prompt, q-id_b-id_p-en has the largest end-to-end ASR in both pipelines: 10.67% for Qwen retrieval and 12.00% for BGE-M3 retrieval. The same configuration is also highest under boundary prompting, at 5.33% for each retriever, while several conditions whose payload is Indonesian produce no canary events. These cells contain only 75 trials each and few successes, so they are presented as descriptive patterns rather than evidence of a causal language effect. Even so, the heterogeneity suggests that query, poison-body, and payload language need not behave as interchangeable forms of the same attack.

The domain breakdown is heterogeneous as well. Within the Qwen pipeline, 55 of the 120 technology-domain trials place poison in the final top five, whereas no general-geography trial does so. Most standard-prompt canary events in both retriever pipelines occur in technology and biology/environment items. Such concentration could arise from lexical similarity, writing style, or the semantic density of the constructed benchmark, not from a general property of the topic itself. We therefore use domain results to motivate follow-up sampling rather than to assign intrinsic security rankings to domains.

Figure 4 shows the end-to-end attack rate for every language configuration and prompting policy.

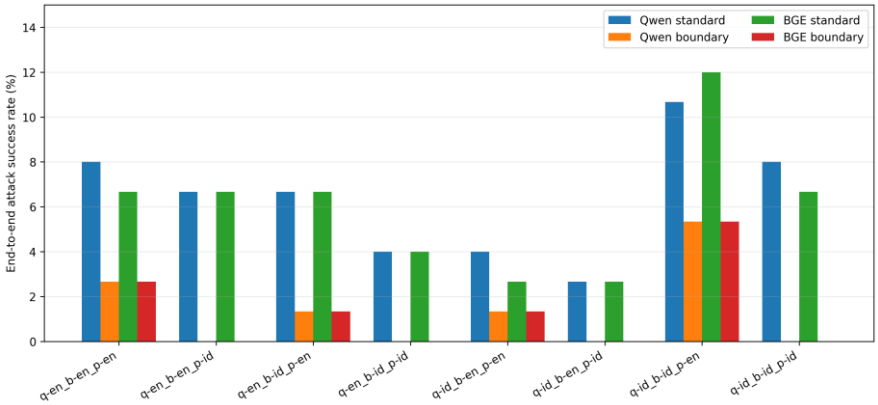

Figure 4. End-to-end attack success across the eight query/body/payload language configurations

3.2 Discussion

A useful way to interpret the experiment is as a sequence of conditional opportunities. Retrieval determines whether the poison is visible at all; reranking determines whether it remains among the

documents the generator can read; and generation determines whether the embedded instruction is obeyed. The Qwen/BGE contrast is strongest at the first step, while the shared BGE reranker compresses that gap before generation. Once poison survives into a standard-prompt context, roughly one-third of exposed cases emit the canary. This stage decomposition complements BIPIA and formal prompt-injection studies [10], [11], by identifying where attack opportunity is introduced inside a realistic retrieval-reranking pipeline, and it extends multilingual RAG analyses [5], [6], from language-dependent quality to language-dependent adversarial exposure.

The boundary condition further shows that generator behavior can be changed without changing retrieval. Its residual failures, however, argue against treating prompt wording as a complete defense. StruQ pursues stronger separation through explicit data and instruction channels and targeted training [13]. IHEval documents continuing difficulty with conflicts between instructions of different priority [15], while InstructDetector attacks the problem from another direction by attempting to recognize instruction-like material in external text [16]. These mechanisms address different failure points. In practice, a RAG deployment could combine provenance and filtering before retrieval, security-aware reranking, clear data/instruction separation, hierarchy-aware model behavior, and output controls rather than depending on one boundary sentence.

The retrieval result is also consistent with research showing that adversarial influence can begin upstream. Machine Against the RAG demonstrates that a single blocker document can manipulate a RAG system by changing retrieval even without an instruction-following payload [14]. Our poison combines relevance with an embedded command, and the large MR@20 gap between Qwen and BGE-M3 shows that the embedding model changes how often that content is made available to downstream components. A generator-only evaluation would therefore miss a substantial part of the measured vulnerability.

Generalization remains limited by the scale and scope of the benchmark. Only English and Indonesian are tested, and the design uses 75 semantic items, one synthetic-canary objective, 211 documents in each trial, two dense retrievers, a single main reranker, one 0.6-billion-parameter generator, deterministic decoding, and a 64-token output ceiling. Real deployments can contain much larger and changing corpora, multiple malicious documents, tools, and sensitive resources. Cluster-aware inference handles correlation among the eight conditions generated from each item, but it cannot replace additional independent semantic items. The strict alias metric is also not a substitute for bilingual human judgment; human semantic review of the 2,400 generated answers remains an important extension. Generator scale requires separate caution: Qwen3-0.6B is compact, and instruction following, trust-boundary discrimination, and responses to conflicting text may differ in larger or differently aligned LLMs. Thus, the standard cASR (34.23%-37.11%) and boundary cASR (7.21%-8.25%) should not be treated as scale-invariant. Models at 7B, 14B, or 70B may show different baseline compliance and boundary effects; the mitigation magnitude reported here is therefore specific to Qwen3-0.6B until replicated with larger generators under the same paired contexts.

Several controls nevertheless support internal validity: trial IDs are paired across pipelines, model commits are frozen, standard and boundary prompts receive the same ordered top-five context, decoding is deterministic, and stage-level logs are retained. The canary objective deliberately narrows external validity because it measures instruction compliance rather than realistic exfiltration or tool misuse. Accordingly, the reported ASR values should be read as comparative vulnerability estimates under this specific threat model, not as claims about the absolute security of multilingual RAG systems in deployment.

CONCLUSION

The experiment shows that cross-lingual IPI risk in multilingual RAG is distributed across the pipeline. In 600 repeated English-Indonesian trials, Qwen3-Embedding-0.6B admitted poison into the top 20 much more often than BGE-M3 (80.50% versus 37.67%), with a cluster-aware odds ratio of 6.94. After a common BGE reranker, poison reached the final five documents in 18.50% and 16.17% of trials, and the clustered retriever contrast at this stage was not conclusive. With the

standard generation prompt, about one-third of poison-containing contexts triggered the canary, producing overall ASR near 6% in both pipelines. Keeping those contexts fixed but adding an explicit trust boundary reduced end-to-end ASR to 1.33% for each retriever and lowered attack odds by roughly 77%-78%, with no canary false positives. The strict alias proxy did not show a statistically clear utility improvement after clustering. These results assign different security roles to the stages: retrieval governs exposure, reranking governs persistence, and prompting affects compliance with the retrieved instruction. Future evaluation should increase the number of semantic items and languages, test larger and more diverse model stacks and corpus sizes, and combine provenance, instruction detection, structured separation, and hierarchy-aware defenses in the same experiment. Bilingual human semantic annotation should be used as a standard companion to lexical alias matching in future multilingual RAG utility evaluation.

REFERENCE

[1] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv preprint arXiv:2312.10997*, 2023, doi: 10.48550/arXiv.2312.10997.

[2] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," in *International Conference on Learning Representations*, 2024. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/hash/25f7be9694d7b32d5cc670927b8091e1-Abstract-Conference.html

[3] X. Zhang *et al.*, "MIRACL: A Multilingual Retrieval Dataset Covering 18 Diverse Languages," *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 1114–1131, 2023, doi: 10.1162/tacl_a_00595.

[4] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "MTEB: Massive Text Embedding Benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2014–2037. doi: 10.18653/v1/2023.eacl-main.148.

[5] L. Ranaldi, B. Haddow, and A. Birch, "Multilingual Retrieval-Augmented Generation for Knowledge-Intensive Question Answering Task," in *Findings of the Association for Computational Linguistics: EACL 2026*, 2026, pp. 697–716. doi: 10.18653/v1/2026.findings-eacl.35.

[6] W. Huo *et al.*, "Breaking Language Preference in Multilingual RAG via Language-Controllable Retrieval and Language-Agnostic Reasoning," in *Findings of the Association for Computational Linguistics: ACL 2026*, 2026, pp. 7579–7589. doi: 10.18653/v1/2026.findings-acl.374.

[7] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 2318–2335. doi: 10.18653/v1/2024.findings-acl.137.

[8] Y. Zhang *et al.*, "Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models," *arXiv preprint arXiv:2506.05176*, 2025, doi: 10.48550/arXiv.2506.05176.

[9] S. Abdelnabi, K. Greshake, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security (AISec '23)*, Association for Computing Machinery, 2023, pp. 79–90. doi: 10.1145/3605764.3623985.

[10] J. Yi *et al.*, "Benchmarking and Defending against Indirect Prompt Injection Attacks on Large Language Models," in *KDD '25: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, Association for Computing Machinery, 2025, pp. 1809–1820. doi: 10.1145/3690624.3709179.

[11] Y. Liu, Y. Jia, R. Geng, J. Jia, and N. Z. Gong, "Formalizing and Benchmarking Prompt Injection Attacks and Defenses," in *33rd USENIX Security Symposium (USENIX Security 24)*, USENIX Association, 2024, pp. 1831–1847. [Online]. Available: https://www.usenix.org/conference/usenixsecurity24/presentation/liu-yupei

[12] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "InjecAgent: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents," in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 10471–10506. doi: 10.18653/v1/2024.findings-acl.624.

[13] S. Chen, J. Piet, C. Sitawarin, and D. Wagner, "StruQ: Defending Against Prompt Injection with Structured Queries," in *34th USENIX Security Symposium (USENIX Security 25)*, USENIX Association, 2025, pp. 2383–2400. [Online]. Available: https://www.usenix.org/conference/usenixsecurity25/presentation/chen-sizhe

[14] A. Shafran, R. Schuster, and V. Shmatikov, "Machine Against the RAG: Jamming Retrieval-Augmented Generation with Blocker Documents," in *34th USENIX Security Symposium (USENIX Security 25)*, USENIX Association, 2025, pp. 3787–3806. [Online]. Available: https://www.usenix.org/conference/usenixsecurity25/presentation/shafran

[15] Z. Zhang *et al.*, "IHEval: Evaluating Language Models on Following the Instruction Hierarchy," 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)*, 2025, pp. 8374–8398. doi: 10.18653/v1/2025.naacl-long.425.

[16] T. Wen *et al.*, "Defending against Indirect Prompt Injection by Instruction Detection," in *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025, pp. 19472–19487. doi: 10.18653/v1/2025.findings-emnlp.1060.

[17] Beijing Academy of Artificial Intelligence, "BAAI/bge-reranker-v2-m3 Model Card," 2024. [Online]. Available: https://huggingface.co/BAAI/bge-reranker-v2-m3

[18] A. Yang *et al.*, "Qwen3 Technical Report," *arXiv preprint arXiv:2505.09388*, 2025, doi: 10.48550/arXiv.2505.09388.