ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

Design and Optimization of a RAG Knowledge Base for Knowledge Management in Chip Testing Enterprises

Linglong Mao*, Shuang Wang

School of Electronic Engineering, Tianjin University of Technology and Education, Tianjin, 300222, China
16638109931@163.com

Abstract: *To improve technical-document retrieval and knowledge reuse in chip testing enterprises, this study develops a retrieval-augmented generation (RAG) knowledge base. Chip models and test-equipment names are linked with document segments, and dense-sparse hybrid retrieval is combined with HyDE query expansion, chip-model/equipment-name filtering, and knowledge-graph retrieval. Experiments on five technical documents and 40 question-answer pairs show that HyDE improves relevant-content recall and name filtering strengthens consistency between answers and source documents, while the effect of knowledge-graph retrieval varies with document structure and question type. The system provides a practical approach to technical-document retrieval in chip testing enterprises.*

Keywords: *chip testing enterprises; knowledge management; retrieval-augmented generation; hybrid retrieval; knowledge graph*

1. Introduction

Chip testing enterprises frequently consult technical documentation during equipment use and testing operations. As the numbers of test instruments, chip models, and documents increase, similar terms may occur across multiple sources, making directory- or keyword-based retrieval prone to inaccurate scope determination and inefficient searching. In practice, user queries also often omit equipment names or chip models and may differ from the standardized terminology used in technical documents, further increasing retrieval difficulty.

Retrieval-Augmented Generation (RAG) retrieves external documents before generating an answer grounded in the retrieved content[1], and its performance depends on the relevance and completeness of the recalled information. HyDE expands query semantics by generating hypothetical text[2]; BGE-M3 supports both dense and sparse representations[3]; RRF is used to fuse results from multiple retrieval paths[4], while a cross-encoder can further rerank the candidates[5]. Knowledge graphs complement text-only similarity retrieval by introducing entity relationships[6]. Recent studies have also treated querying, retrieval, generation, and evaluation as consecutive stages of a RAG system[10-12].

To meet the knowledge-management needs associated with technical materials in chip testing enterprises, this study constructs a RAG knowledge base built on hybrid retrieval and incorporates HyDE, chip-model and equipment-name filtering, and knowledge-graph retrieval. Experiments are conducted on five technical documents and 40 question-answer pairs, with module ablation and grouped comparisons used to examine the contribution of each method. Retrieval performance is evaluated using Recall@k and MRR, while answer and context quality are assessed with the RAGAS framework [7].

2. System Design

The system consists of knowledge-base construction and online question answering. The former performs document parsing, chunking, chip-model or equipment-name recognition, vectorization, and database ingestion, and can optionally construct a knowledge graph. The latter rewrites the query, performs text retrieval and optional graph retrieval, fuses and reranks the results, and then uses a large language model to generate the answer. All three optimization modules are implemented as

ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

configurable switches to enable comparison under identical conditions.

2.1 Retrieval Optimization Methods

1) HyDE Query Expansion

HyDE (Hypothetical Document Embeddings) is used to mitigate mismatches between user expressions and the terminology used in technical documents[2]. The system first generates a hypothetical answer related to the query and then uses both the hypothetical answer and the original query for retrieval; the final answer remains grounded in the retrieved source text.

2) Chip-Model and Equipment-Name Filtering

Chip-model and equipment-name filtering is used to constrain the retrieval scope. The system identifies the target model or equipment name from the current query and conversation history and filters document chunks through the item_name field in Milvus[8]; when the target cannot be determined, retrieval is performed over the entire document collection. In the ablation experiments, USE_ENTITY_FILTER controls only this field-level filtering, while model/equipment recognition and query rewriting remain unchanged.

3) Knowledge Graph Augmentation

The knowledge graph organizes entities and their relationships in the documents and preserves the correspondence between entities and source-text chunks, thereby supplementing associated information that may not be directly captured by text retrieval.

During construction, a large language model extracts entities and relations. After name normalization, deduplication, and endpoint checking, relations with confidence scores greater than 0.3 are retained. Entities and relations are written to Neo4j, entity-name vectors are stored in Milvus, and entities are linked to source-text chunks through MENTIONED_IN relations.

During retrieval, entity names are first extracted from the query, excluding any confirmed chip models or equipment names. Graph entities are then matched using dense and sparse scores with weights of 0.4 and 0.6, respectively, followed by one-hop neighborhood retrieval. Source-text chunks associated with the matched entities are added to the candidate set, while relation information is simultaneously provided to the large language model.

2.2 End-to-End Workflow Design

1) Knowledge Base Construction Workflow

The knowledge base sequentially performs document parsing, body-text processing, chunking, chip-model or equipment-name recognition, vectorization, and database ingestion. Knowledge-graph construction is connected as an optional branch after vectorization, as shown in Fig. 1.

The workflow is managed using LangGraph. PDF files are converted to Markdown with MinerU[9], while native Markdown documents proceed directly to body-text processing; extracted images are stored in MinIO. The body text is first segmented by headings, after which long passages are recursively split using a target length of 600 characters and an overlap of 50 characters. Segments shorter than 400 characters may be merged with adjacent content, with the merged length capped at 1,000 characters.

After chunking, chip models or equipment names are identified, and BGE-M3 is used to generate 1,024-dimensional dense vectors and sparse vectors for storage in Milvus. When the knowledge graph is enabled, entities and relations are additionally extracted and written to Neo4j. The chunking, vector, and graph parameters remain identical across all experimental groups; only the on/off states of the optimization modules are varied.

2) Knowledge Base Retrieval Workflow

For online question answering, the query is first rewritten using the current question, conversation history, and any identified chip model or equipment name. Both conventional retrieval and HyDE retrieval use dense-sparse hybrid retrieval with weights of 0.7 and 0.3, respectively, and each path retains the top 20 chunks. Results from multiple paths are fused using RRF with a path weight of 1.0 and a smoothing parameter of 60; after deduplication, the top 20 candidates are retained.

ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

Candidate chunks are reranked using a cross-encoder. Chunks exceeding 512 tokens are compressed according to the existing system rules before scoring, while answer generation still uses the original text; 2-6 chunks are retained after reranking. When the knowledge graph is enabled, the related entities, relations, and corresponding source text returned by graph retrieval are also added to the candidate set, after which the large language model generates the final answer.

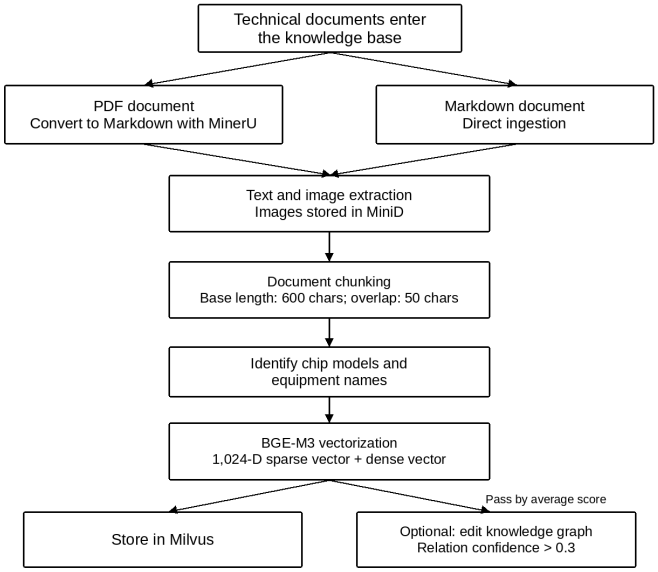


Fig. 1 Knowledge base construction workflow

2.3 Experimental Setup

1) Dataset

In the main experiment, five technical documents, including ATE test-equipment documentation, chip datasheets, and chip test-result reports, are ingested into the knowledge base, yielding 300 text chunks and 2,250 graph entities. Based on these materials, 40 Chinese question-answer pairs are prepared. The questions cover chip models, equipment information, technical parameters, and operating instructions, and the reference answers are taken from the source documents. For each question, the reference_contexts supporting the answer are also stored and used as the relevant-chunk set for retrieval evaluation.

The 40 question-answer pairs are divided into two groups of 20 according to whether the question explicitly includes a chip model or equipment name. Model names, series names, and equipment names are all treated as valid identifiers and recorded by has_subject. The detailed distribution is shown in Table 1.

Table 1 Dataset statistics

Document	Questions	With chip/equipment identifier	Without chip/equipment identifier
V93000.pdf	9	5	4
Chroma 3380 VLSI.pdf	10	5	5
ADS8517.pdf	6	3	3
V93000_20260708_0820_Orion-A7_V93-2607-4100_WF01-A1.pdf	6	3	3
V93000_20260712_1127_Nebula-P3_V93-2607-4101_WF02-A2.pdf	9	4	5
Total	40	20	20

2) Evaluation Metrics

Retrieval performance is evaluated using Recall@k and mean reciprocal rank (MRR). Let Q denote the question set; for a question q, let Gq denote the set of relevant chunks, Rkq the top-k retrieval results, and rankq the rank of the first relevant chunk.

$$\text{Recall@}k = \frac{1}{|Q|}\sum_{q\in Q}\frac{|G_q \cap R_q^k|}{|G_q|} \quad (1)$$

ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

Recall@k measures the coverage of relevant chunks among the top-k retrieval results. This study calculates k = 1, 3, and 5, and reports RRF Recall@5 as the primary retrieval result.

$$\mathrm{MRR} = \frac{1}{|Q|}\sum_{q\in Q}\frac{1}{\mathrm{rank}_q} \tag{2}$$

Answer and context quality are evaluated using the RAGAS (Retrieval Augmented Generation Assessment) framework[7]. RAGAS uses large language models to assist in evaluating retrieved contexts and generated answers, with different metrics computed from the question, retrieved context, generated answer, and reference information. This study uses faithfulness, context precision, context recall, answer correctness, factual correctness, and noise sensitivity. Noise sensitivity reflects the extent to which generated answers are affected by irrelevant retrieved content, with lower values indicating less susceptibility to noise. Detailed definitions and calculation methods are provided in Ref. [7]. The evaluation model is qwen-flash, and semantic computation uses BGE-M3 dense vectors.

3) Evaluation Protocol

Module ablation experiments are used to examine the independent contribution of each optimization module. The full configuration, in which all optimization modules are enabled, serves as the reference. One module is disabled at a time, and the resulting changes in each metric are compared to determine the module's contribution to system performance. Five configurations are included in the main experiment: the baseline configuration disables HyDE, chip-model and equipment-name filtering, and the knowledge graph; the full configuration enables all three; the other three configurations are derived from the full configuration by disabling one module at a time. The baseline and full configurations are also compared separately for questions with and without chip/equipment identifiers.

All five configurations use the same knowledge base and question set, with BGE-M3, the generation model, the reranking model, and conversational conditions held constant; web search is disabled. In the baseline configuration, USE_HYDE, USE_ENTITY_FILTER, and USE_KG are all set to 0, whereas all three are set to 1 in the full configuration. Each ablation configuration disables one of these options, while all other retrieval parameters remain unchanged. The detailed settings are shown in Table 2.

Table 2 Five configurations used in the main experiment

System configuration	HyDE	Chip/equipment name filtering	Knowledge graph
Base configuration	Shut down	Shut down	Disabled
Full configuration	Open	Open	Enabled
Shut down HyDE	Shut down	Open	Enabled
Shut down name filtering	Open	Shut down	Open
Shut down knowledge graph	Open	Open	Shut down

3. Experimental Results and Discussion

3.1 Module Ablation Results

Following the ablation design described above, the full configuration is used as the reference, and chip-model/equipment-name filtering, HyDE, and the knowledge graph are disabled individually, producing retrieval and answer-quality metrics for five configurations. The retrieval results are shown in Table 3, and the context and answer evaluation results are shown in Table 4. In the tables, “Enabled” indicates that the module is active in the full configuration, whereas “Disabled” indicates the result obtained after disabling that module while keeping the rest of the full configuration unchanged.

Table 3 Retrieval results under different module settings

Module compared	RRF Recall@5		Rerank MRR	
	Open	Disabled	Enabled	Disabled
Chip/device name filtering	0.540	0.510	0.908	0.850
HyDE	0.540	0.495	0.908	0.846
Knowledge graph	0.540	0.660	0.908	0.921

ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

Table 4 Context and answer evaluation results under different module settings

Module status	Faithfulness	Context Recall	Answer Correctness	Factual Correctness	Noise Sensitivity↓
Chip/device name filtering Open	0.892	0.925	0.793	0.694	0.164
Chip/device name filteringShut down	0.806	0.913	0.746	0.622	0.123
HyDEOpen	0.892	0.925	0.793	0.694	0.164
HyDEShut down	0.853	0.875	0.721	0.675	0.168
Knowledge graph Open	0.892	0.925	0.793	0.694	0.164
Knowledge graph Shut down	0.955	0.975	0.841	0.666	0.175

1) Effect of Chip-Model and Equipment-Name Filtering

Enabling chip-model and equipment-name filtering increases faithfulness from 0.806 to 0.892, factual correctness from 0.622 to 0.694, and MRR from 0.850 to 0.908, while noise sensitivity increases from 0.123 to 0.164. These results indicate that name filtering narrows the retrieval scope and improves the main answer-quality metrics, although different evaluation dimensions do not change in the same direction.

2) Effect of HyDE Query Expansion

With HyDE enabled, context recall increases from 0.875 to 0.925, answer correctness from 0.721 to 0.793, and MRR from 0.846 to 0.908. The results indicate that enriching query semantics helps retrieve more source content related to chip models, test equipment, and operating parameters.

3) Effect of Knowledge Graph Augmentation

After enabling the knowledge graph, RRF Recall@5 decreases from 0.660 to 0.540 and MRR from 0.921 to 0.908; faithfulness, context recall, and answer correctness also decrease. In contrast, factual correctness increases from 0.666 to 0.694, while noise sensitivity decreases from 0.175 to 0.164.

Graph retrieval adds one-hop neighboring entities and their associated source text to the candidate set. When a query requires relational information, such evidence may complement text retrieval; however, when relations are sparse or too many entities are expanded, the composition of the candidate set may also be altered. The main experiment shows that the knowledge graph does not improve all metrics simultaneously, and its effectiveness depends on entity matching, relation structure, and question type.

3.2 Grouped Analysis by Chip-Model and Equipment Identifier

The baseline and full configurations are evaluated on two groups of questions, with and without chip/equipment identifiers, with 20 questions in each group. The results are shown in Table 5.

Table 5 Comparison of the full and baseline configurations by chip-model and equipment identifier

Metric	With chip/equipment identifier		Without chip/equipment identifier	
	integrity	Baseline	Full	Baseline
RRF Recall@5	0.450	0.490	0.630	0.540
Rerank MRR	0.842	0.842	0.975	0.875
Degree of fidelity	0.835	0.938	0.949	0.919
Context accuracy	0.829	0.762	0.925	0.931
Context recall	0.875	0.900	0.975	0.850
Answer correctness	0.751	0.797	0.835	0.718
Factual correctness	0.574	0.650	0.814	0.671

Among the 20 questions that do not explicitly specify a chip model or equipment name, context recall increases from 0.850 to 0.975, answer correctness from 0.718 to 0.835, factual correctness from 0.671 to 0.814, and MRR from 0.875 to 0.975; context precision decreases slightly from 0.931 to 0.925. These results show that the full configuration primarily improves retrieval coverage and answer quality when the query contains incomplete identifying information.

Among the 20 questions that explicitly specify a chip model or equipment name, context precision increases from 0.762 to 0.829 and MRR remains at 0.842, whereas faithfulness and factual correctness decrease from 0.938 and 0.650 to 0.835 and 0.574, respectively. Thus, when the query already provides

ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

a clear model or equipment name, adding retrieval branches does not yield consistent gains.

3.3 Supplementary Validation of the Knowledge Graph

To further examine the role of the knowledge graph in information retrieval that requires cross-entity associations, an additional multi-entity, multi-relation general software-service question-answering set is used for supplementary validation. Only the baseline RAG and knowledge-graph-augmented configurations are compared, as shown in Table 6. This supplementary experiment is used to compare the behavior of the knowledge-graph module under different question structures.

Table 6 Context and answer evaluation results in the supplementary experiment

System configuration	Faithfulness	Context recall	Answer correctness	Factual correctness
Base RAG(baseline)	0.876	0.875	0.764	0.655
Knowledge graph enhancement	0.912	0.924	0.819	0.735

In the supplementary validation, adding the knowledge graph increases faithfulness, context recall, answer correctness, and factual correctness from 0.876, 0.875, 0.764, and 0.655 to 0.912, 0.924, 0.819, and 0.735, respectively.

Compared with the main experiment, the supplementary validation yields more consistent improvements on multi-entity relational questions, indicating that knowledge graphs are more suitable for queries requiring associations among multiple entities; their benefits are less stable for questions concerning an explicit model or a single parameter.

4. Conclusion

This study constructs a RAG-based technical-document knowledge base for chip testing enterprises. Building on hybrid retrieval, the system incorporates HyDE, chip-model and equipment-name filtering, and knowledge-graph retrieval, and evaluates these components through module ablation and grouped experiments.

The main experiment on five technical documents and 40 question-answer pairs shows that HyDE improves context recall, answer correctness, and MRR; chip-model and equipment-name filtering improves faithfulness, factual correctness, and MRR, although noise sensitivity increases. The improvements of the full configuration are concentrated primarily in questions that do not explicitly specify a chip model or equipment name.

The knowledge graph improves only some metrics in the main experiment, whereas all four metrics increase in the supplementary validation involving multi-entity relational questions. These results suggest that graph retrieval is better suited to queries involving richer relational information and should not be used as a fixed retrieval branch for every query.

In practical deployment, the correspondence among chip models, test-equipment names, and technical documents can first be standardized, after which HyDE, name filtering, and knowledge-graph retrieval can be selected according to common query types. Future work can incorporate additional real-world device documentation and evaluate response latency, invocation cost, and storage overhead.

References

[1] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks[C]//Advances in Neural Information Processing Systems 33. 2020: 9459-9474.

[2] Gao L, Ma X, Lin J, et al. Precise zero-shot dense retrieval without relevance labels[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2023: 1762-1777.

[3] Chen J, Xiao S, Zhang P, et al. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation[C]//Findings of the Association for Computational Linguistics: ACL 2024. Bangkok: Association for Computational Linguistics, 2024: 2318-2335. DOI: 10.18653/v1/2024.findings-acl.137.

[4] Cormack G V, Clarke C L A, Büttcher S. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods[C]//Proceedings of the 32nd International ACM SIGIR Conference

ISSN 2616-5775 Vol. 9, Issue 9: 40-46, DOI: 10.25236/AJCIS.2026.090906

on Research and Development in Information Retrieval. 2009: 758-759.
[5] Nogueira R, Cho K. Passage re-ranking with BERT[EB/OL]. (2019-01-13). https://arxiv.org/abs/1901.04085.
[6] Edge D, Trinh H, Cheng N, et al. From local to global: A graph RAG approach to query-focused summarization[EB/OL]. (2024-04-24). https://arxiv.org/abs/2404.16130.
[7] Es S, James J, Espinosa-Anke L, et al. RAGAs: Automated evaluation of retrieval augmented generation[C]//Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024: 150-158.
[8] Wang J, Yi X, Guo R, et al. Milvus: A purpose-built vector data management system[C]//Proceedings of the 2021 International Conference on Management of Data. 2021: 2614-2627.
[9] Wang B, Xu C, Zhao X, et al. MinerU: An open-source solution for precise document content extraction[EB/OL]. (2024-09-27). https://arxiv.org/abs/2409.18839.
[10] Ni B, Liu Z, Wang L, et al. Towards trustworthy retrieval augmented generation for large language models: A survey[EB/OL]. (2025-02-08). https://arxiv.org/abs/2502.06872.
[11] Gan A, Yu H, Zhang K, et al. Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey[EB/OL]. (2025-04-21). https://arxiv.org/abs/2504.14891.
[12] Li Y, Zhang W, Yang Y, et al. A survey of RAG-reasoning systems in large language models[C]//Findings of the Association for Computational Linguistics: EMNLP 2025. Suzhou: Association for Computational Linguistics, 2025: 12120-12145. DOI: 10.18653/v1/2025.findings-emnlp.648.