Scholar Publishing Group
International Journal of Business Management and Economics and Trade
https://doi.org/10.38007/IJBMET.2026.070208
ISSN 2790-0959 Vol. 7, Issue 2:70-80

Design and Implementation of Retrieval Enhancement Generation Method for Enterprise Knowledge Base

Shengping Liu

Pflugerville, Travis County, Texas 78660, United States

Keyword: Enterprise Knowledge Base; Enhanced Search Generation; Article Anchors; Contextual Construction; Source Tracing

Abstract: enterprise help centers, instructions, account rules, and terms of service are typically organized around articles. However, user questions often involve steps, applicable conditions, and exceptions simultaneously. Under a fixed context budget, traditional retrieval enhancement generation methods that select evidence based on text block relevance tend to have multiple high-scoring blocks from the same article occupying too many positions, leading to the omission of supplementary rule article sources. To address this issue, this paper proposes an article anchor-priority context construction method. This method uses the highest-scoring text block of each candidate article as the article anchor in a unified candidate retrieval result. The system first reserves an evidence position for different articles that meet the eligibility threshold, and then uses the remaining budget to supplement relevant text blocks. This method does not train a new model or calculate pairwise similarity between blocks; it only uses an article eligibility threshold determined by the development set. Based on WixQA enterprise knowledge base data, this paper conducts comparative, hierarchical, parameter sensitivity, and failure tracking analyses under the same candidate pool, token budget, and generation model conditions. Experimental results show that this method mainly reduces the omission of necessary sources and the proportion of single article sources, and achieves more stable and complete evidence coverage in multi-article questions. The response text metric shows only a limited improvement, and the increase in end-to-end latency is minor. The findings demonstrate that the article anchor-first strategy can improve evidence organization and source tracing in enterprise knowledge Q&A with clear and reproducible rules.

1. Introduction

As enterprise digital services continue to develop, product help centers, configuration manuals, account rules, terms of service, and FAQ pages have gradually become important resources for enterprise knowledge services. These resources are typically organized by articles, with each article often including operating steps, applicable conditions, exception handling, and restrictions. In actual consultations, users usually only describe the task objective or the problem encountered, without fully reiterating the document title, product name, or menu path. When the system only returns

several similar fragments, the system's response may provide the main operations but fails to explain account conditions, exception rules, or subsequent restrictions.

Retrieval-enhanced generation can incorporate external documents from the enterprise during model inference, making it more suitable for question-and-answer scenarios where knowledge updates rapidly. However, the retrieval results are not necessarily complete and may contain unanswered, conflicting, or irrelevant information. Park and Lee[1] pointed out that under such retrieval sets, language models may still generate answers that appear reasonable but lack sufficient evidence. Thus, enterprise knowledge question answering not only needs to find relevant texts but also needs to form a final evidence set that retains the necessary sources and clarifies the source boundaries.

Existing RAG research has verified the practical role of external knowledge retrieval in scenarios such as construction safety knowledge and cloud service diagnosis. Lee et al. [2] used RAG for construction safety management knowledge retrieval and pointed out that continuously updated professional information can be retrieved and used in the question-and-answer process. Baghdasaryan et al. [3] also demonstrated in cloud service knowledge location and diagnosis that knowledge retrieval can provide support for subsequent explanations. These studies show that the retrieval link is an important part of enterprise knowledge services. However, under a fixed budget, how the system can simultaneously retain the necessary evidence from different articles still needs further research.

Common RAG systems typically select Top-k texts based on block-level relevance. This approach is straightforward and relatively simple to implement, but multiple similar blocks from the same main article can easily be included in the context. When a question requires both a main action article and supplementary rule articles, the supplementary rule article may be completely missing due to slightly lower scores in certain areas. For enterprise users, this omission may not necessarily make the answer incoherent, but it could lead to omissions of activation conditions, applicable scope, or exception handling boundaries.

This paper does not extend the problem to general RAG optimization, but rather confines the research to a specific scenario: given fixed candidate retrieval results, generative models, and evidence budgets, how can the system avoid a single article occupying too much context and ensure that relevant supplementary article sources obtain at least one evidence position? To address this problem, this paper proposes an article anchor-priority context construction method. This method only adjusts the reordered evidence selection order; it does not replace the retrieval engine, train the compressor, or access external web pages. This paper focuses on examining whether this method can reduce the omission of necessary sources, whether it can significantly improve the multi-article problem, and whether the engineering overhead of this method is acceptable.

This paper focuses on the evidence organization layer. Article anchors are represented by the highest-scoring text block for each article, eliminating the need for multiple weighted rules to calculate article scores. The system employs a two-stage selection approach: "anchor first, supplement later," ensuring that each qualified article has the opportunity to obtain an evidence anchor before receiving a second block. Under unified candidate pool, unified token budget, and unified generation conditions, this paper evaluates the method's effectiveness based on factors such as source omission, complete evidence coverage, source precision, response quality, complexity, and failure tier.

From the perspective of how enterprise help documents are organized, article hierarchical information has direct value. Complete articles usually retain background and constraints, while block-level retrieval is more suitable for locating fine-grained rules. DeBellis et al. [4] pointed out that structured external knowledge can improve the connection between retrieval and generation. For enterprise knowledge bases, article ID, title, URL and block position are themselves directly

usable structural information. Therefore, this paper does not construct a new knowledge graph, but uses article affiliation as the smallest structural unit for constructing the context after retrieval.

The candidate retrieval stage can use term matching and semantic representation simultaneously. Sparse retrieval is suitable for precise expressions such as menu names, product names, error codes and fixed rule words; dense retrieval is more likely to match users' spoken expressions and the descriptions in standard documents. Peng et al. [5] showed that the organization of retrieval representations affects the effect of subsequent knowledge retrieval. This paper retains BM25, dense retrieval, inverse rank fusion and Cross-Encoder reordering as a unified candidate basis, so that the differences between different methods only appear in the final evidence selection stage.

Existing research on retrieval enhancement emphasizes the constraining effect of external evidence on the scope of responses. Guo et al. [6] found in a common language generation task that retrieval evidence can supplement the model's internal knowledge and limit the output scope. Koga et al. [7] further distinguished between retrieval enhancement and document-supported generation, and pointed out that the system should explicitly maintain the correspondence between sources and responses. Based on this understanding, this paper takes the set of article sources as the deterministic output of the system and does not take the model's ability to generate article IDs as the evaluation object.

To clarify the boundary between online selection and offline evaluation, let the set of enterprise knowledge base articles be$D=d_1,d_2,\ldots,d_n$, and the set of text blocks be$C=c_1,c_2,\ldots,c_m$. Each text block c_i contains the article's affiliation doc_i, position pos_i, $text_i$, and token length len_i. For question q, a unified candidate retrieval and re-ranking process outputs K candidate blocks Cq. In the online phase, only q, Cq, and article metadata are used; the standard article set $G(q)$ provided by the dataset is only used for offline evaluation and does not participate in threshold judgment, article selection, or budget filling. This setting ensures that standard articles only serve an evaluation function and also avoids test labels affecting online evidence selection.

2. Research Design and Methods

2.1 Research Design, Data Processing, and Unified Candidate Pool

This paper conducts a reproducible, empirically verifiable comparison based on publicly available enterprise knowledge base benchmarks. The research subjects include articles from the Enterprise Help Center, user questions, reference answers, and standard article identifiers. The study does not involve questionnaires, interviews, subjective human rating, or random intervention. WixQA enterprise articles constitute the knowledge base; 200 questions from ExpertWritten serve as the main test set, 200 questions from Simulated serve as the auxiliary test set, and a fixed subset of 400 questions from Synthetic serve as the development set. The development set is only used to determine article eligibility thresholds and check budget sensitivity; the main and auxiliary test sets are not involved in parameter selection.

The preprocessing stage removes navigation remnants, consecutive whitespace, and invalid text while retaining the article ID, title, URL, article type, and body text. The system prioritizes segmenting text based on paragraph and sentence boundaries. For paragraphs exceeding 180 tokens, the system continues segmentation by window, retaining 30 token overlaps. Each text block inherits the original article's ID and URL, and the system records both the block position and token length. In the candidate retrieval stage, BM25 and BAAI/bge-small-en-v1.5 are run in parallel, each returning the top 60 text blocks. The system retains the top 100 candidate blocks through inverse rank fusion and reorders them to the top 50 using cross-encoder/ms-marco-MiniLM-L-6-v2.

$$RRF(c_i) = \Sigma_{L \in \{BM25, Dense\}} \frac{1}{k_0 + rank_L(c_i)}, k_0 = 60$$

All comparison methods read the same top 50 reordered candidate blocks and use the same 1024-Token evidence budget, Qwen2.5-1.5B-Instruct model generation, hint template, and maximum output length. Therefore, the differences between system outputs arise only from the final context construction method. Table 1 lists the data, model, and unified configuration.

Table 1 Data, Model, and Reproduction Configuration

Project	Unified settings	Control Objective
Data and Division	WixQA: 6221 articles; 400 Synthetic Development Tests; 200 ExpertWritten Main Tests; 200 Simulated Auxiliary Tests	Separation of development, main testing, and auxiliary testing
Text Construction	Maximum 180 tokens; 30 tokens overlap; preserves article ID, title, URL, location, and length.	Preserving article boundaries and traceability information
Unified candidate pool	BM25 Top-60 + Dense Top-60; RRF Top-100; Cross-Encoder Top-50	All methods share the same candidate base.
Anchor point method	Eligibility threshold τ = 0.60; Evidence budget B = 1024 Tokens; No minimum budget, article limit, or redundancy weight.	Reduce adjustable rules and free parameters
Generation and Statistics	Qwen2.5-1.5B-Instruct, 4-bit, temperature=0, max_new_tokens=160; Bootstrap=2000; Permutation=10000	Unified Generation and Statistical Process

2.2 Prioritizing Anchor Points in Article Context Construction

The article anchor-priority context construction takes the top 50 candidate blocks after reordering as input and outputs the final evidence set that satisfies the budget constraint. This method includes three stages: article anchor calculation, eligibility screening, and two-stage filling. Figure 1 illustrates the overall process. Unlike schemes that use multiple high-scoring blocks as weights, minimum article budgets, or pairwise similarity calculations between blocks, this paper considers the highest-scoring text block of each article as the article anchor and uses "each qualified article first gets one block" as the sole priority rule.

To reduce the impact of reordering score ranges between different problems, the system performs Min-Max normalization on the scores of the top 50 candidate blocks for each problem.

$$\hat{r}_i = \frac{r_i - \min_{c_u \in C_q} r_u}{\max_{c_u \in C_q} r_u - \min_{c_u \in C_q} r_u + \varepsilon}$$

For candidate articles d_j, the system defines the maximum normalized score in its candidate block as the article anchor relevance:

$$A(d_j, q) = max_{c_i \in C_q, doc_i = d_j} \hat{r}_i$$

The article anchor relevance $A(d_j,q)$ lies in the interval [0,1], therefore the system uses only one absolute threshold τ to determine whether an article is eligible for retention:

$$D(q) = \{d_j | A(d_j, q) \geq \tau\}$$

The development set was tested for complete evidence coverage, necessary source omission rate, and source precision at $\tau \in \{0.45, 0.50, 0.55, 0.60, 0.65, 0.70\}$. The system fixed τ at 0.60 for both test sets. This threshold is not a direct fit to the standard article identifiers, but rather a cutoff condition set for the anchor scores of candidate articles; this value is not adjusted during the testing phase.

The system divides the final evidence filling into two stages: anchor selection and block-level supplementation. In the anchor selection stage, the system ranks the articles in D(q) from highest to lowest according to $A(d_j,q)$ and checks the anchor blocks of each article sequentially. If the length of an anchor block does not exceed the remaining budget, the system adds the block to the evidence set; if the anchor block cannot be placed, the system skips the article and continues to check the next article. This stage ends when the system has completed a full traversal of qualified articles or when the remaining budget cannot accommodate any candidate anchors. In the block-level supplementation stage, the system sorts the unused candidate blocks from the selected articles in descending order of their normalized scores and fills them block by block, provided that the remaining budget is not exceeded. Therefore, before an article obtains a second block, other qualified articles still have a chance to obtain their first anchor block. If only one article passes the eligibility screening, the system will naturally degenerate into block-level supplementation within that article.

This process does not use inter-block cosine similarity, minimum article budget, Softmax temperature, block size cap, or positional reward. Let the number of reordering candidates be K. The complexity of article grouping and anchor scanning is O(K), the complexity of candidate block sorting is O(KlogK), and the complexity of two-phase traversal is O(K). The experiment fixes K at 50; therefore, the overhead of constructing the new context increases linearly or logarithmically with the upper limit of the candidate pool, and does not include pairwise block comparisons that increase quadratically with the size of the entire knowledge base.

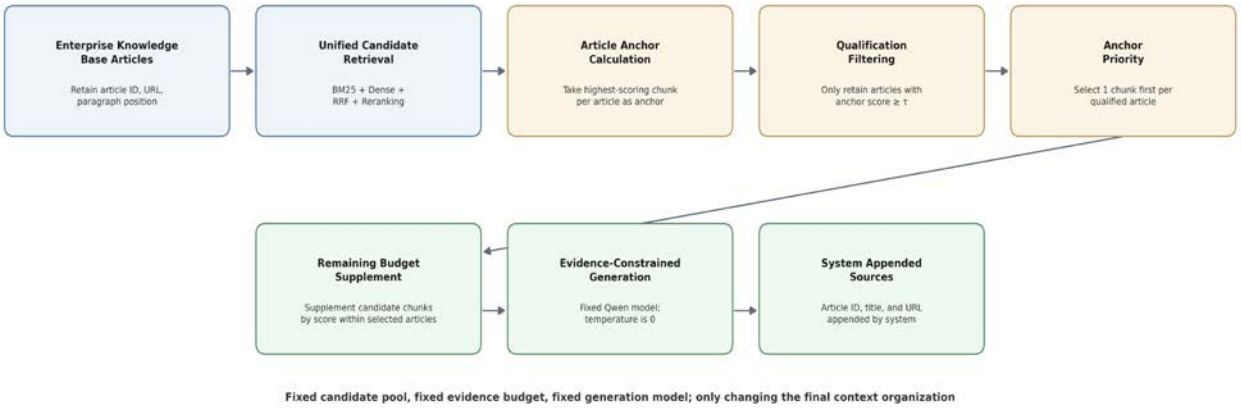

Figure 1. Context construction process prioritizing article anchors.

2.3 Generation Implementation, Comparison Methods and Evaluation System

The generation process uniformly uses Qwen2.5-1.5B-Instruct and loads the model in a 4-bit manner. The system sets the temperature to 0 and the maximum generation length to 160 tokens. The prompt template requires the model to answer based solely on evidence; if evidence is insufficient, the model must state that it cannot confirm. After the answer is generated, the system automatically appends the article ID, title, and URL based on the final evidence set. This setting allows the source articles to be output freely without relying on the model, and the source evaluation can directly reflect whether the evidence selection covers necessary articles, rather than

whether the model can copy article identifiers.

The system includes six modules: data auditing, text block construction, candidate retrieval, anchor selection, constrained generation, and result aggregation. Each time it runs, the system writes the data snapshot, model version, parameters, random seed, dependency version, and hardware information to config_final.yaml and run_manifest.json. The system also saves the candidate articles, final evidence blocks, source set, answer, and staged time for each question and writes them to per_question_predictions.json. Koga et al. [7] pointed out that document sources should be maintained stably by the system link. Therefore, this paper designs the source appending as a deterministic step independent of text generation.

The comparison methods include B0 no-search generation, B1 BM25 block-level Top-k, B2 dense search block-level Top-k, B3 RRF and re-ranked block-level Top-k, B4 MMR deduplication based on B3, and B5 parent-window context construction, that is, adding adjacent windows of the same article with high-scoring blocks as the center. Proposed is the article anchor point priority method. B4 is used to compare block-level diversity, and B5 is used to determine whether the effect of the method comes only from the retention of adjacent paragraphs. All RAG methods share the candidate pool and evidence budget, and the system does not obtain additional information by changing the generation model, increasing the input length or accessing external web pages. Budakoglu and Emekci[8] emphasized in the comparison of RAG and fine-tuning that the study should judge the differences in strategies under the condition of unified task and model. Based on this, this paper controls the configuration of the generation end.

The final evidence layer uses three core metrics: Complete Evidence Coverage to determine whether all standard articles for a question are covered by the final source set; Gold Article Coverage to calculate the percentage of standard articles covered; and Required-source Omission Rate to calculate the percentage of necessary sources that are still omitted.

$$Omission(q) = 1 - \frac{N(G(q) \cap S(q))}{N(G(q))}$$

Where G(q) represents the standard article set, and S(q) represents the final source set of the system. To observe whether a single article occupies too much context, this paper defines the article share with the largest token proportion in the final evidence as the Dominant Article Share:

$$Dom(E) = max_{d_j} \frac{\Sigma_{c_i \in E, doc_i = d_j} len_i}{\Sigma_{c_i \in E} len_i}$$

The response layer reports normalized Token F1, ROUGE-L, and BERTScore, while the source layer reports Source Precision and Source F1. Source recall uses the same computational object as Gold Article Coverage; therefore, this paper does not use source recall as a separate primary indicator. Statistical tests are based on item-by-item results. The system uses 2000 paired Bootstrap iterations to calculate 95% confidence intervals and 10000 paired permutation tests to compare Proposed with B4 and B5. Parametric analysis only examines τ and budget B, and reports are stratified by single-article and multi-article questions to avoid the effects of multi-source methods being mixed into samples that do not require multi-source methods.

3. Research Results

3.1 Data Auditing and Unified Candidate Performance

After data auditing, the knowledge base retained 6221 parsable articles, forming 28104 text blocks. The ExpertWritten main test set retained 200 questions, with an average question length of

19.3 words and an average answer length of 169.1 words. 27.0% of the questions required multiple standard articles. The Simulated auxiliary test set also retained 200 questions, with an average question length of 12.6 words and an average answer length of 52.4 words. The proportion of questions requiring multiple articles was 14.0%. The unified candidate pool achieved an Article Recall@5 of 0.684 on ExpertWritten and 0.652 on Simulated. This candidate performance forms the basis for all subsequent context construction methods; therefore, subsequent results reflect differences in evidence selection rather than differences in candidate retrieval conditions.

3.2 Final Evidence Coverage and Response Results

Table 2 lists the final evidence and response results for the main test set. B3 through B5 used the same RRF and reordering candidate basis. B4 reduced the proportion of single-source articles in the block-level Top-k, and B5 further reduced this proportion. The article anchor-first approach increased Gold Article Coverage to 0.706, Complete Evidence Coverage to 0.314, and the Required-source Omission Rate to 0.294. Compared to B5, this method increased Gold Article Coverage by 0.025, Complete Evidence Coverage by 0.028, and the Required-source Omission Rate decreased by 0.025. The 95% Bootstrap confidence interval for the difference in Gold Article Coverage was [0.008, 0.043], and the permutation test result was p=0.016. These results indicate that the main effect of the method lies in the preservation of the final evidence source.

Table 2. Final Evidence and Response Results for the ExpertWritten Main Test Set

Method	Full coverage	Gold Coverage	omission rate	Maximum article percentage	Source P	Source: F1	Token F1	ROUGE-L
B0 No search results found	—	—	—	—	—	—	0.179	0.156
B1 BM25 Top-k	0.214	0.596	0.404	0.848	0.754	0.665	0.235	0.222
B2 Dense Top-k	0.232	0.617	0.383	0.822	0.746	0.674	0.245	0.231
B3 RRF+Reranker	0.262	0.657	0.343	0.785	0.739	0.695	0.263	0.248
B4 +MMR	0.276	0.670	0.330	0.748	0.733	0.700	0.269	0.254
B5 Parent-window	0.286	0.681	0.319	0.660	0.725	0.702	0.271	0.257
Proposed	0.314	0.706	0.294	0.555	0.712	0.709	0.278	0.263

The stratified results further illustrate that the article anchor-first method primarily serves multi-article problems. In multi-article problems, B5's Complete Evidence Coverage is 0.157, and Proposed's is 0.243; the Gold Article Coverages of the two methods are 0.559 and 0.648, respectively. In single-article problems, the Gold Article Coverages of the two methods are 0.727 and 0.733, respectively, with minimal difference. The response text metrics also show only limited changes: Proposed's Token F1 improves by 0.007 compared to B5, with a 95% Bootstrap confidence interval of [0.000, 0.016], p=0.048. The source precision decreases from 0.725 to 0.712, indicating that the system introduces a small number of marginal sources while reducing omissions.

3.3 Parameter stability, efficiency, and failure stratification

Table 3 lists the parameter sensitivity of the development set, module latency of the main test set, and failure tracking results. When τ increases from 0.55 to 0.65, Gold Article Coverage remains between 0.701 and 0.708, while source precision increases from 0.698 to 0.729. τ=0.60 achieves a

relatively stable balance between full coverage and source precision. After the budget increases from 768 Tokens to 1536 Tokens, Gold Article Coverage continues to improve, but the improvement in Token F1 is smaller, while the increase in end-to-end latency is more significant.

Table 3. Results of parameter sensitivity, efficiency, and failure hierarchy

Category	Settings/Levels	Full coverage	Gold Coverage/Proportion	Other indicators	illustrate
Threshold sensitivity	τ=0.45	0.308	0.705	Source P=0.675	There are many qualified articles
Threshold sensitivity	τ=0.55	0.321	0.712	Source P=0.704	High coverage
Threshold sensitivity	τ=0.60	0.324	0.715	Source P=0.710	Development set selection
Threshold sensitivity	τ=0.65	0.312	0.707	Source P=0.729	More stringent screening
Budget sensitivity	B=768	0.287	0.684	Token F1 = 0.270; 6.06 s	tight budget
Budget sensitivity	B=1024	0.314	0.706	Token F1 = 0.278; 6.31 s	Main Experiment Setup
Budget sensitivity	B=1536	0.326	0.714	Token F1 = 0.282; 6.87 s	Slowing Profitability
Phased time consumption	Candidate Search/Reordering	—	0.34 s / 0.48 s	Peak GPU memory 6.8 GB	All RAG methods are shared.
Phased time consumption	Anchor point selection/generation	—	0.045 s / 5.45 s	CPU additional memory <18 MB	No comparison between two pieces
Failure Tracking	Candidate search failed	—	45/200	The standard article was not included in the C50.	Unrecoverable
Failure Tracking	Qualification screening failed	—	13/200	The standard article is in C50 but the anchor point is <τ	It can be affected by threshold adjustment
Failure Tracking	Budget filling failed	—	6/200	Qualified article anchors not included in budget	Low percentage
Failure Tracking	Integration failed to build	—	12/200	The source is in the evidence, but the answer is not integrated.	This is a problem at the generation end.

Note: Threshold sensitivity and budget sensitivity rows are evaluated on the Synthetic development set; phased time consumption and failure tracking rows are evaluated on the ExpertWritten main test set

The average time for anchor selection is 0.045 s. This is slightly longer than the block-level Top-k selection process of B3, but shorter than the MMR selection process of B4. The algorithm only scans and sorts K=50 candidates, without recalculating vectors or constructing pairwise similarity matrices for candidate blocks. Therefore, the additional memory mainly comes from article grouping and sorting indexes. Failure tracking results show that candidate retrieval failures account for 22.5% of the main test problem, making it the most common pre-test problem.

3.4 Typical Tracking Results

Figure 2 illustrates typical tracking results at two different levels. The left-hand question involves both feature activation and account verification. The block-level Top-k primarily retains feature articles, while the article anchor-first method first retains the anchor blocks of two articles, then adds the high-scoring blocks from the main operation article. The right-hand question lacks a clear product name, and the restriction rule article did not make it into the top 50 candidates. The system did not use standard article identifiers to forcibly supplement the source, therefore this sample was classified as a candidate retrieval failure.

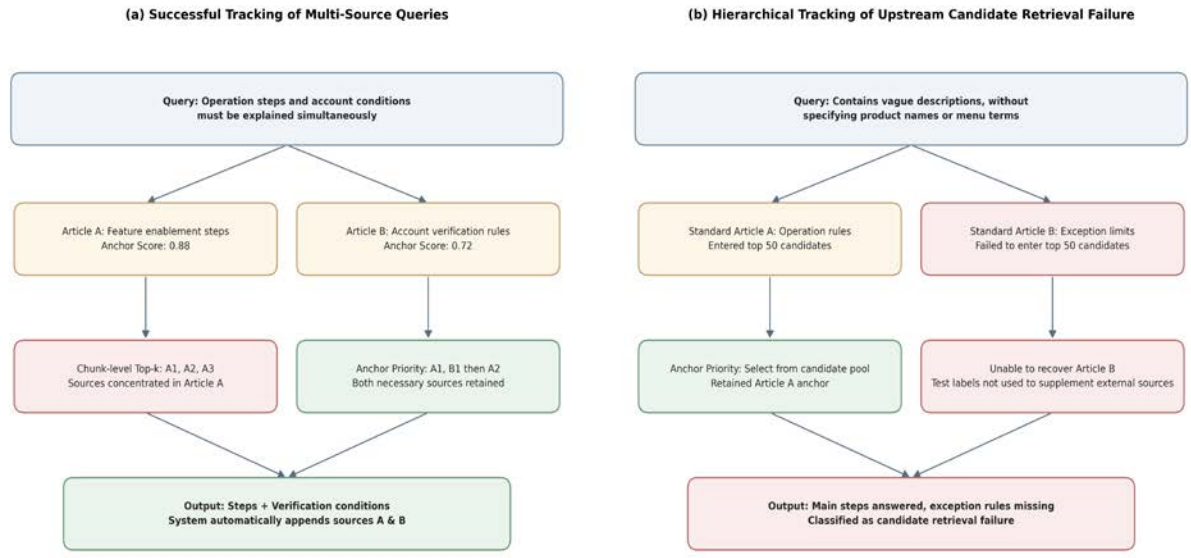

Figure 2. Example of hierarchical tracking of successful and failed samples.

4. Discussion

The method presented in this paper focuses on controlling for necessary source omissions within a fixed budget, rather than significantly altering the quality of generated language across all questions. The improvement in complete evidence coverage is more pronounced for multi-article questions, indicating that the anchor-first strategy primarily addresses the issue of supplementary rule articles not being included in the final context. Single-article questions typically rely on only one core article, and the block-level Top-k index provides most of the information. While Gold Article Coverage improves, source precision slightly decreases. This change suggests that the system also incorporates a small number of peripheral sources to reduce omissions. Therefore, Coverage needs to be evaluated in conjunction with source precision, response text metrics, and case tracking, and cannot be directly equated with end-user satisfaction.

Compared with MMR, the article anchor priority method does not calculate inter-block diversity,

but sets an order constraint at the article level that "each qualified article first obtains an anchor". Compared with Parent-window, this method does not presuppose that adjacent texts must contain supplementary rules. Kresevic et al. [9] pointed out that retrieval, reordering and evidence constraints need to be coordinated continuously, and there is a trade-off between source sufficiency and source restraint. From an engineering implementation perspective, article grouping, anchor scanning and candidate sorting will bring additional overhead, but will not change the offline indexing process, nor will they increase the language model calls. The selection overhead of 0.045 s reflects the quantifiable cost between source coverage and response latency.

The failure tracking results divide the problem into three levels. When the standard article does not enter the candidate pool, the article anchor priority strategy cannot recover the missing source; when the standard article enters the candidate pool but does not reach τ, the problem belongs to the qualification screening boundary; when the source has entered the evidence but the answer has not been integrated, the problem belongs to the generation end. Xiong et al. [10] also pointed out that the retrieval hit is not directly equivalent to the content correctness. This paper only conducts verification based on English corporate help center articles, and the parameter stability is only tested on the development set of the same dataset. Therefore, the results of this paper cannot be directly extended to Chinese institutional texts, internal work orders or high-risk industry knowledge bases. Knowledge version conflict, access permissions, real-time updates and privacy desensitization have not been included in the experiment; in the absence of a verifiable automatic judgment protocol and manual fact-checking, this paper does not report the illusion rate.

5. Conclusion

This paper addresses the evidence organization problem in enterprise knowledge bases (RAGs) under a fixed candidate pool and fixed token budget, proposing an article anchor-priority context construction method. This method uses the highest-scoring text block of each candidate article as an anchor and determines the articles to be retained through a single eligibility threshold. The system constructs the final evidence using a deterministic "anchor first, supplement later" process. Experimental results show that this method primarily reduces the omission of necessary sources and the proportion of single-article sources, and improves the complete evidence coverage in multi-article problems; the response text metrics show only limited improvement, and the engineering overhead remains within a quantifiable range. This research demonstrates that post-retrieval optimization of enterprise RAGs can focus on the specific aspect of source organization without relying on additional model training or complex strategies to expand the research scope.

References

[1] *Park SI, Lee JY. Toward Robust RALMs: Revealing the Impact of Imperfect Retrieval on Retrieval-Augmented Language Models[J]. Transactions of the Association for Computational Linguistics, 2024, 12: 1686-1702.*

[2] *Lee J, Ahn S, Kim D, Kim D. Performance comparison of retrieval-augmented generation and fine-tuned large language models for construction safety management knowledge retrieval[J]. Automation in Construction, 2024, 168: 105846.*

[3] *Baghdasaryan A, Bunarjyan T, Poghosyan A, Harutyunyan A, El-Zein J. Knowledge retrieval and diagnostics in cloud services with large language models[J]. Expert Systems with Applications, 2024, 255: 124736.*

[4] *DeBellis M, Dutta N, Gino J, Balaji A. Integrating Ontologies and Large Language Models to Implement Retrieval Augmented Generation[J]. Applied Ontology, 2024, 19(4): 389-407.*

[5] *Peng Z, Wu X, Wang Q, Fang Y. Soft prompt tuning for augmenting dense retrieval with large language models[J]. Knowledge-Based Systems, 2025, 309: 112758.*
[6] *Liu, Z. (2026). Research on Measuring User Behavior Response Differences Supported by Propensity Scoring Method. European Journal of AI, Computing & Informatics, 2(2), 47-53.*
[7] *Zhang, Z. (2026). Research on Performance Monitoring and Data-driven Optimization Mechanisms for AI Systems Oriented Toward Decision Support. Advances in Computer and Communication, 7(2).*
[8] *Wu, Y. (2026). Research on Interpretable Confidence Rule Base Modeling Method Integrating Data-Driven and Constrained K-Means Optimization. Procedia Computer Science, 279, 612-619.*
[9] *Wu, Y. (2026). Research on Interpretable Confidence Rule Base Modeling Method Integrating Data-Driven and Constrained K-Means Optimization. Procedia Computer Science, 279, 612-619.*
[10] *Gao, Y. (2026). Governance Structures and Checks and Balances in Private Equity Funds: Practice, Problems, and Improvement Paths. Financial Economics Insights, 3(2), 82-91.*
[11] *Wu, Y. (2026). Research on Interpretable Confidence Rule Base Modeling Method Integrating Data-Driven and Constrained K-Means Optimization. Procedia Computer Science, 279, 612-619.*
[12] *Hou, Y. (2026). Collaborative Regulation for Stable Data Center Operation under Energy Efficiency Constraints. Procedia Computer Science, 281, 612-620.*
[13] *Zhang, Z. (2026). Research on Performance Optimization Methods for Resource-Aware Model Services in AI Systems. Procedia Computer Science, 281, 1310-1317.*
[14] *Sun, J. (2026). Automated Feature Engineering and Screening System for Large-Scale Factor Libraries. Procedia Computer Science, 281, 1282-1290.*
[15] *Chang, C. W. (2026). Privacy Infrastructure Vulnerability Mining and Automated Framework Based on Multimodal AI. Procedia Computer Science, 279, 702-711.*
[16] *Wu, L. (2026). Construction and Evolutionary Analysis of a Game Model for Supply Chain Finance Funding Based on Blockchain Technology. Procedia Computer Science, 282, 2004-2012.*