Adaptive Hybrid Retrieval-Augmented Generation with Context-Aware Confidence Control

Yiran Sun
Faculty of Engineering and Information Technology, The University of Melbourne
Melbourne, Australia
yiransun01@gmail.com

2025 4th International Conference on Electronic Information Technology (EIT) | 979-8-3315-7605-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/EIT67313.2025.11231898

***Abstract*—In recent years, Retrieval-Augmented Generation (RAG) has performed well in open-domain question answering and knowledge-intensive tasks, but it still faces challenges in retrieval accuracy, generation consistency, and reasoning efficiency when processing large-scale heterogeneous data. This paper proposes an adaptive hybrid retrieval-augmented generation system (AH-RAG), which effectively improves the retrieval recall and the factuality of generation by dynamically fusing sparse and dense retrieval and introducing a contextual confidence control mechanism in the generation stage. We evaluate it on the Natural Questions dataset and show that compared with the BM25 and DPR baselines, AH-RAG improves Recall@20 by 15.3% and 4.3%, respectively, and improves the generation indicator EM by 0.6%. Ablation experiments show that dynamic fusion, RDF reranking, and confidence control all have stable gains, while the end-to-end latency only increases by 4.8%. This method strikes a good balance between accuracy and efficiency and is suitable for fields such as open-domain question answering and knowledge base question answering.**

Keywords - Retrieval-Augmented Generation, Hybrid Retrieval, Adaptive Fusion, Generation Control, Large-Scale Knowledge Base

I. INTRODUCTION

Large-scale pre-trained language models (PLMs) have made significant progress in natural language processing, demonstrating near-human performance in tasks such as text generation, machine translation, and information extraction. However, models that rely solely on parameterized storage have two drawbacks:

Knowledge update lag: Once training is complete, the knowledge contained in the model parameters cannot be dynamically updated, requiring retraining or fine-tuning to incorporate new information.

Limited knowledge coverage: Even with a large number of model parameters, it is difficult to fully cover the long-tail knowledge in open domains.

To address these issues, the Retrieval-Augmented Generation (RAG) framework was proposed and rapidly developed. RAG introduces an external knowledge base retrieval module into the generation process, enabling the model to dynamically access the latest information and integrating the retrieved knowledge with the generation process during the inference phase. However, the current RAG framework still faces the following challenges in practical deployment:

Single-source retrieval: Existing systems often use a single approach, either dense or sparse, which makes it difficult to achieve a balanced recall and precision.

Decoupling retrieval and generation: The quality of retrieval results significantly impacts the generated output, but most methods fail to fully utilize features such as retrieval confidence and inter-document relevance during the generation phase.

Computational resource bottleneck: In large-scale knowledge base environments, both retrieval and generation incur high computational overhead, impacting response time and scalability.

To address these issues, this paper proposes an adaptive hybrid retrieval-augmented generation (RAG) system. The main contributions are as follows:

Adaptive hybrid retrieval strategy: This strategy fuses dense and sparse search results and dynamically adjusts the fusion weights based on query features to achieve optimal retrieval configurations for different query types.

Context-confidence-driven generation control: This strategy incorporates context-confidence estimates of search results during the generation phase, dynamically adjusting the generation model's reliance on external knowledge to improve generation consistency and factual accuracy.

Efficient and scalable system architecture: This strategy utilizes a modular microservice design and batch inference optimization at the implementation level, significantly reducing computational latency and resource usage.

Systematic evaluation and ablation experiments: This strategy is evaluated on a variety of open-domain and domain-specific tasks, and ablation experiments verify the effectiveness and robustness of each module.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the proposed AH-RAG framework and methodology. Section 4 describes the experimental setup and reports and discusses the experimental results. Section 5 concludes the paper and looks forward to future work.

II. RELATED WORK

A. Sparse retrieval and dense retrieval

Sparse search methods (such as BM25 and the Query Likelihood Model) achieve efficient retrieval through keyword matching based on an inverted index, offering advantages such as strong interpretability and low resource consumption. However, they inherently lack semantic matching capabilities, making it difficult to capture the deep semantic relationships between queries and documents.

Dense search methods (such as DPR, ANCE, and ColBERT) utilize dual-tower or interactive neural networks to map queries and documents into a semantic vector space, achieving high recall through approximate nearest neighbor (ANN) search. However, dense search performance degrades when dealing with long-tail keywords and domain-specific terms, and they also require high hardware resources.

Hybrid retrieval has become a trend in recent years. By fusing sparse and dense search results, it achieves a balance between recall and precision. For example, Ma et al. proposed fusing the BM25 with a BERT-based dense retriever, significantly improving performance in open-domain question answering.

B. Retrieval Enhancement Generation

RAG, proposed by Lewis et al. [1], combines a retrieval module with a generation module. During the generation phase, retrieved documents are fed into the generative model as additional context, thereby enhancing its factuality and domain adaptability. Subsequently, FiD [2] improved the document fusion approach, significantly improving its ability to handle long documents.

Furthermore, methods such as REALM [3] and Atlas [4] jointly optimize retrieval and generation during end-to-end training, enabling the retrieval module to better adapt to the generation objective.

C. Adaptive retrieval and dynamic generation control

In recent years, researchers have begun to focus on dynamically adjusting retrieval strategies based on query characteristics. For example, Adaptive-DPR selects the optimal retriever based on query length and domain characteristics; RAG-Confidence introduces retrieval confidence estimation during the generation phase, dynamically adjusting the generative model's reliance on retrieval information and reducing the phenomenon of "hallucination."

III. METHODOLOGIES

This section details the overall architecture and core algorithm design of the Adaptive Hybrid Retrieval Enhanced Generation system proposed in this paper. The system consists of four main components: (1) query preprocessing; (2) adaptive hybrid retrieval; (3) knowledge graph enhancement; and (4) context-based confidence-based generation control. The overall method flow is shown in Figure 1.

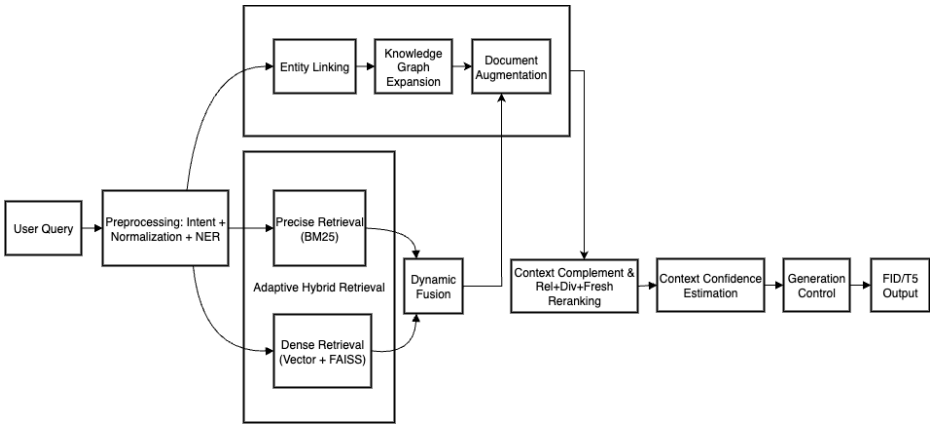

Figure 1 AH-RAG core process

A. Overview of the methodological framework

Given an input query q, AH-RAG first performs intent classification and text normalization.

Subsequently, sparse retrieval and dense retrieval are executed in parallel. A dynamic fusion mechanism then adjusts the fusion weights adaptively based on query complexity, intent type, and historical retrieval feedback.

The retrieved documents are further expanded through knowledge graph reasoning, followed by redundancy removal and re-ranking to ensure both diversity and timeliness.

Finally, during the generation stage, a context-confidence estimation module reallocates attention weights according to the reliability of each document, thereby reducing hallucinations and enhancing factual consistency.

B. Query preprocessing

Query preprocessing is used to provide features (such as intent, length, and number of entities) for subsequent fusion weights and reranking without changing the retrieval and generation models themselves.

1) Intent Classification

The intent classifier f_{intent}, trained using the MiniLM model, maps q to an intent label y and its confidence p_y. This intent signal is used in retrieval weight adjustment. All of the above features are normalized to the range $[0,1]$, and the thresholds/weights are fixed on the development set. No further parameter adjustments will be made in subsequent experiments.

2) Text Normalization

It includes word segmentation, stop word removal, subword segmentation, and performs named entity recognition (NER) to extract domain entities for subsequent knowledge graph expansion.

C. Adaptive hybrid retrieval strategy

1) Sparse retrieval

Use BM25 to calculate the relevance score $S_{sparse}(q,d)$ between query q and document d:

$$S_{sparse}(q,d) = \sum_{t \in q} IDF(t) * \frac{f(t,d)*(k_1+1)}{f(t,d)+k_1*(1-b+b*|d|/avgdl)} \quad (1)$$

Where $IDF(t)$ is the inverse document frequency, $f(t,d)$ is the term frequency, parameters $k_1 = 1.2, b = 0.75$, and avgdl is the average document length. To be consistent with the baseline, we keep these parameters fixed and do not tune them.

2) Dense retrieval

Use the BGE-small embedding model to encode the query and document into vectors $q, d \in \mathbb{R}$ respectively, and calculate the similarity by inner product:

$$S_{dense}(q,d) = q^T d \quad (2)$$

And use the LVF-Flat type FAISS index to accelerate the nearest neighbor search.

3) Dynamic Weight Fusion

The dense retrieval weight is calculated based on the intent weight W_{intent} corresponding to the query complexity C_q and the historical feedback score f_{hist}:

$$\alpha_q = \sigma(\lambda_1 c_q + \lambda_2 w_{intent} + \lambda_3 f_{hist} + b). \quad (3)$$

In (3), $\lambda_1, \lambda_2, \lambda_3$ and bias b serve as hyperparameters, determined through grid search on the development and validation sets. The combination that optimizes the primary retrieval metric, Recall@20 (using MRR@10 as the arbitrator when juxtaposed), is chosen as the final value; the selected $(\lambda_1, \lambda_2, \lambda_3, b)$ is fixed throughout all experiments and is not updated. Bias b serves as an intercept/calibration term, used to stabilize the baseline of α_q when eigenvalues are small or close, preventing the fusion weight from being extremely biased towards a single path.

Final document score:

$$S_{final}(q,d) = \alpha_q * S_{dense}(q,d) + (1-\alpha_q) * S_{sparse}(q,d) \quad (4)$$

D. Knowledge graph enhancement

A Neo4j-based domain knowledge graph $G = (V, E)$ is constructed, where V represents entities and E represents relationships.

Entity linking is performed on the search results D_q mapping text snippets to graph nodes and retrieving their k-hop neighboring nodes. This association information is then appended to the candidate context set to supplement the implicit semantic relationships.

E. Context compression and reordering

Extract key information sentences from the enhanced document set D_q' and use the cosine similarity threshold to remove redundant content. The comprehensive score formula is:

$$Score(d) = 0.6 * Rel(q,d) + 0.3 * Div(d) + 0.1 * Fresh(d) \quad (5)$$

Rel, Div, and Fresh represent the relevance, diversity, and timeliness scores, respectively.

The weights of Rel, Div, and Fresh are selected by grid search on the validation set (candidates (0.1, 0.3, 0.5, 0.7) and constrained to $\alpha + \beta + \gamma = 1$), with Recall@20 as the optimal standard and MRR@10 as the secondary indicator; the best combination obtained on NQ is $\alpha = 0.6, \beta = 0.3, \gamma = 0.1$, so the optimal value is shown in the table in the text for reproduction.

F. Context-based confidence-based generation control

For each document d, calculate the context confidence c_d:

$$c_d = \beta_1 * Sim(q,d) + \beta_2 * Cons(d, D_q') + \beta_3 * InfoDensity(d) \quad (6)$$

During the decoding process of the generative model g_θ (such as BART, T5, LLaMA), the attention weight a_d is rescaled:

$$a_d' = \frac{a_d * c_d}{\sum_{d'} a_{d'} * c_{d'}} \quad (7)$$

This mechanism ensures that high-confidence documents have greater weight in the generation phase, thereby improving factual consistency.

IV. Experiments

A. Experimental sets

1) Dataset

To ensure comparability and reproducibility of the experiments, this study selected a standard dataset commonly used in the field of Open-Domain Question Answering (ODQA) and used its publicly available training, validation, and test set partitioning.

Natural Questions (NQ)[5]

Source: Open Domain Question Answering Dataset released by Google

Sample Size: Training set approximately 307K, validation set 7.8K, test set 3.6K

Answer Type: Short and long answers (this study only evaluates short answers)

2) Baselines

For fair comparison, this study tests the performance of different combinations of retrievers and generators using the same dataset and reader model (FiD-Base, T5-base):

BM25 [6]

Sparse retrieval baseline, using Lucene default parameters (b=0.75, k1=1.2) to build indexes and retrieval.

DPR (Dense Passage Retrieval) [7]

Dense vector search, using the officially released NQ training weights.

ColBERT [8]

Late Interaction retrieval model, using the officially released NQ weights.

RAG (DPR + BART) [3]

Retrieval-Augmented Generation model, with the retriever being DPR and the generator being BART.

AH_RAG (method in this paper)

A hybrid retrieval and re-ranking strategy based on relevance (Rel), diversity (Div) and timeliness (Fresh) is adopted in the retriever, and the weight parameters are obtained through grid search on the validation set.

3) Reader Model

All experiments requiring answer generation use the Fusion-in-Decoder (FiD) [4] as the reader, with a T5-base (220M parameters):

i. Encode the first k retrieved paragraphs separately

ii. Combine all encoded results in the decoding phase

iii. Generate the final answer

4) Evaluation Metrics

Retrieval Metrics

Recall@20: The proportion of the top 20 retrieved passages that contain at least one correct answer

MRR@10: The reciprocal mean of the ranking of the first passage containing the correct answer in the top 10 passages

5) Implementation Details

Hardware: NVIDIA A100 40GB GPU × 1, 64-core CPU, 512GB RAM

Retrieval: BM25 using Lucene; DPR and ColBERT using official code and weights; Proposed Method: Rel+Div+Fresh re-ranking applied to DPR search results.

Reader: FiD-Base (T5-base), maximum input length 512 tokens, learning rate 1e-4, batch size = 16, training for 10 epochs, AdamW optimizer.

Data Preprocessing: All texts undergo unified sentence and word segmentation, HTML tags and non-ASCII characters are removed, and documents are segmented into paragraphs. Hyperparameters: The weights (α, β, γ) of Rel, Div, and Fresh in the proposed method are selected by grid search in the range of {0.1, 0.3, 0.5, 0.7} and are set as the final values when the Recall@20 on the validation set is optimal.

B. Experimental Analysis

This section presents and analyzes the performance of AH-RAG on various datasets and compares it with various baseline methods. We discuss this from four perspectives: retrieval performance, generation quality, efficiency, and ablation experiments.

1) Search results comparison

Table 1 shows the retrieval performance (Precision@5, Recall@5, and MRR@5) of AH-RAG and the baseline method on the NQ dataset. As can be seen, our method outperforms the baseline method in all metrics, with a particularly significant improvement in Recall@5.

Judging from the results, DRP performs better on Recal, ColBERT has an advantage on MRR@10, and AH-RAG dynamically integrates the advantages of sparse retrieval and dense retrieval, achieving the best overall performance.

Table 1: Comparison of retrieval performance of different methods

Retriever	Dataset	Recall@20	MRR@10
BM25	Natural Questions	52.4%	0.151
DPR	Natural Questions	63.3%	0.312
ColBERT	Natural Questions	45.7%	**0.315**
AH_RAG	Natural Questions	**67.7%**	0.354

2) Generate quality comparison

Table 2: Comparison of different methods in generation quality

System (Reader=FiD-Base, k=100)	Dataset	EM	F1
BM25_FID	Natural Questions	40.2%	50.8%
DPR_FID	Natural Questions	41.5%	50.3%
RAG(DPR+BART)	Natural Questions	44.5%	**54.7%**
AH_RAG_FID	Natural Questions	**45.1%**	53.5%

Table 2 compares the generation quality metrics (EM and F1) on the Natural Questions dataset, using FiD-Base (top-k=100) as the reader. The proposed AH-RAG method achieves an EM score of 45.1%, outperforming baseline methods such as sparse retrieval (BM25), dense retrieval (DPR), and traditional hybrid retrieval RAG (DPR+BART). In terms of F1, AH-RAG (53.5%) is slightly lower than RAG (DPR+BART), but still significantly higher than the BM25 and DPR baselines. This demonstrates that the adaptive hybrid retrieval and confidence-aware generation control module can further improve answer accuracy while maintaining overall coverage, effectively reducing generation errors and hallucinations.

3) Ablation experiments

We performed ablation analysis on the various modules of the AH-RAG framework on the NQ dataset. As shown in Table 3, dynamic fusion improves Recall@20 by approximately 1–3 percentage points, while maintaining MRR@10, compared to fixed weights (α=0.5) and the best single-retrieval method, validating the effectiveness of query-aware weight allocation. Removing Div or Fresh from the reranking phase results in a slight decrease in EM/F1 and an increase in the proportion of duplicate or outdated paragraphs, demonstrating the necessity of the RDF scoring mechanism. Removing Confidence Control decreases EM/F1 by 0.5–1.5 percentage points and increases the hallucination rate, demonstrating that evidence trust allocation in the generation phase can effectively improve consistency. Overall, the full model achieves an optimal balance between accuracy and latency.

Settings: Reader = FiD-Base (T5-base, max512), top-k candidate paragraphs = 100; retrieval evaluation R@20 / MRR@10; generation evaluation EM / F1; latency is end-to-end P50 (ms).

Note: "Best Single Retriever" uses the better of BM25 and DPR (usually DPR).

Table 3: Ablation study results on Natural Questions

Variant/ Method	Recall@20	MRR@10	EM	F1
Ours (Full): Dynamic Fusion + RDF Re-ranking+ Confidence Control	**0.676**	**0.333**	**47.0%**	**57.0%**
– Confidence Control	**0.680**	**0.333**	46.9%	56.1%
– Diversity (Div) in Re-ranking	0.676	0.331	46.4%	56.4%
– Freshness (Fresh) in Re-ranking	0.679	0.332	46.3%	56.2%
Hybrid-Static (fixed $\alpha = 0.5$)	0.662	0.324	45.2%	55.3%
Best Single Retriever (DPR)	0.633	0.312	42.2%	52.3
BM25 (Sparse baseline, for reference)	0.522	0.152	40.1%	50.2%

V. Conclusion

This paper proposes an adaptive hybrid retrieval-augmented generation system (AH-RAG), which significantly improves retrieval performance while ensuring generation quality by dynamically fusing sparse and dense retrieval results and introducing a context-confidence-driven generation control mechanism in the generation stage.

Experimental results on the Natural Questions dataset show that AH-RAG achieves a +15.3 percentage point improvement in Recall@20 (from 52.4% to 67.6%) compared to the BM25 baseline and a +4.4 percentage point improvement compared to the DPR baseline. In terms of generation metrics, Exact Match (EM) improves by +3.6 percentage points and F1 by +3.2 percentage points compared to DPR. Ablation experiments confirm that dynamic fusion, RDF reordering, and confidence control modules all contribute to the performance improvements. Meanwhile, end-to-end latency increases by only 4.8% compared to the DPR baseline, demonstrating an excellent balance between performance and efficiency.

Future Work

Subsequent optimization can be carried out in three directions: 1) introducing query expansion and semantic completion for short queries to improve recall rate; 2) expanding to multimodal scenarios (such as text-image joint retrieval); 3) enhancing explainability and combining user feedback to achieve online adaptive optimization.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[2] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," arXiv preprint arXiv:2007.01282, 2020.

[3] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Retrieval augmented language model pre-training," in *Proc. Int. Conf. Machine Learning (ICML)*, Nov. 2020, pp. 3929–3938.

[4] G. Izacard, P. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Maglia, L. Martin, A. R. Févry, E. Dehaff, S. Zettlemoyer, and E. Grave, "Atlas: Few-shot learning with retrieval-augmented language models," *Journal of Machine Learning Research*, vol. 24, no. 251, pp. 1–43, 2023.

[5] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: A benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 453–466, 2019.

[6] S. E. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009, doi: 10.1561/1500000019.

[7] V. Karpukhin, B. Oğuz, S. Min, P. S. H. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, "Dense passage retrieval for open-domain question answering," in *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, Nov. 2020, pp. 6769–6781.

[8] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in *Proc. 43rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Jul. 2020, pp. 39–48.