Proceedings of the 3rd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2025.27800

Quality Prediction of RAG System Retrieval Based on Machine Learning Algorithms

Chaoyi Yu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
2387663175@qq.com

Abstract. The Retrieval Enhanced Generation (RAG) system improves the accuracy and reliability of content generation by retrieving external knowledge, and has been widely used in fields such as intelligent question answering and knowledge assistants. However, its core performance depends on the quality of the retrieval stage, and the relevance and factual consistency of the retrieval results directly determine the effectiveness of the generated content. However, factors such as query complexity, document noise, and domain differences in real-world scenarios can easily lead to fluctuations in retrieval quality. Traditional manual evaluation is costly and outdated, making it difficult to meet real-time optimization requirements. At the same time, existing models have limitations in complex feature fusion and parameter optimization. Therefore, this article proposes a retrieval quality prediction model that combines the Lizard Optimization Algorithm (HLOA), Convolutional Neural Network (CNN), and Bidirectional Gated Recurrent Unit (BIGRU). Correlation analysis shows that there is a strong positive correlation between retrieval rank and retrieval usefulness score, meaning that the higher the retrieval rank, the better the retrieval usefulness score; The query complexity is strongly negatively correlated with the retrieval usefulness score, meaning that the higher the query complexity, the lower the retrieval usefulness score. Integrate this model with decision trees, random forests Adaboost, The comparison of nine models, including gradient boosting tree, ExtraTrees, CatBoost, XGBoost, LightGBM, and KNN, showed that their performance was overall better: MSE (28.617), RMSE (5.349), MAE (4.401), and MAPE (17.355) were the lowest, while R^2 (0.952) was the highest. This study provides an effective solution for accurate prediction and real-time optimization of the retrieval quality of RAG systems, helping to enhance the application value of RAG technology in practical scenarios.

Keywords: RAG, Horn lizard optimization algorithm, integrated neural network, bidirectional gated recurrent unit, retrieval quality.

1. Introduction

Retrieval enhanced generation (RAG) systems improve the accuracy and reliability of generated content by retrieving external knowledge, and have been widely used in fields such as intelligent question answering and knowledge assistants. Its core performance depends on the quality of the

DOI: 10.54254/2755-2721/2025.27800

retrieval stage - the relevance and factual consistency of the retrieval results directly determine the effectiveness of the generated content. However, in real-world scenarios, factors such as query complexity, document noise, and domain differences often lead to fluctuations in retrieval quality. Traditional manual evaluation is costly and outdated, making it difficult to meet real-time optimization requirements [2]. Therefore, building an efficient retrieval quality prediction model has become the key to the implementation of the RAG system. It is necessary to quantify the matching degree between retrieval results and queries, document credibility, and other indicators to predict the retrieval effect in advance, providing a basis for adjustments in the subsequent generation process.

Machine learning algorithms provide an automated solution for predicting the quality of RAG retrieval [3]. Traditional models achieve preliminary prediction of retrieval relevance by mining manual features such as keyword overlap and entity coverage; Deep learning models can automatically extract deep semantic features from text, capture implicit associations between queries and documents, and significantly improve prediction accuracy. CNN can capture local semantic matching patterns, while RNN models can handle sequence dependencies, effectively handling long texts and complex sentence structures. These algorithms reduce the reliance on artificial feature engineering and can dynamically adapt to data from different fields, providing decision support for real-time optimization of RAG systems. However, there are still problems such as incomplete feature extraction and insufficient model parameter optimization [4].

In response to the limitations of existing models in complex feature fusion and parameter optimization, this paper proposes a prediction model that combines the Lizard Optimization Algorithm (HLOA), Convolutional Neural Network (CNN), and Bidirectional Gated Recurrent Unit (BIGRU). CNN is used to extract local key features between queries and documents, capturing matching patterns at the keyword and phrase levels; BIGRU models bidirectional contextual dependencies of text sequences, enhancing the capture of long-range semantic associations; HLOA optimizes the key parameters of CNN and BIGRU by simulating the global optimization ability of horned lizard predation behavior, avoiding the model from getting stuck in local optima. The collaboration of the three can fully integrate local and global features, improve the fitting ability of complex patterns in retrieval quality, and provide a new solution for accurate prediction of retrieval quality in RAG systems.

2. Data sources

This dataset is mainly used for quality prediction retrieval in RAG systems, consisting of 754 records, 20 independent variables, and 1 predictor variable. The independent variables belong to four types of features: query features involve query complexity, length, ambiguity, type, and domain; The document features include length, number of days to date, authority, format, factual information, noise level, and structural complexity; Matching features include relevance score, semantic similarity, keyword overlap percentage, entity coverage, domain matching degree, and language consistency; The retrieval features include ranking and retrieval methods in the retrieval results. The predictor variable is the retrieval usefulness score. This dataset can be used to train machine learning models to predict the actual level of assistance that retrieval results can provide for generating tasks, providing support for optimizing the performance of RAG systems. Table 1 shows the statistical results of the dataset.

DOI: 10.54254/2755-2721/2025.27800

Table 1. The statistical results of the dataset

Variable Name	Mean	Variance	Median
query_complexity	5.401857	8.578005	5
query_length	16.037135	6.295479	16
query_ambiguity	0.506777	8.523993	0.515
query_type	3.018568	2.012935	3
query_domain	5.404509	8.177456	5
document_length	1029.218833	2.998369	1001.5
document_age_days	1844.591512	1.096582	1826.5
document_authority	0.486671	8.079143	0.485
document_format	2.961538	2.021095	3
document_factuality	0.639416	4.267403	0.63
document_noise	0.352573	4.020825	0.36
document_structural_complexity	2.891247	2.057214	3
relevance_score	0.517308	8.236977	0.53
semantic_similarity	0.502586	8.772836	0.495
keyword_overlap	48.888727	8.065791	50.45
entity_coverage	0.515491	8.090155	0.51
domain_match	0.488992	8.362395	0.48
language_consistency	0.746711	2.087629	0.74
retrieval_rank	10.159151	3.461474	10
retrieval_method	2.48939	1.259515	2.5
retrieval_usefulness_score	41.673873	6.092095	42.1

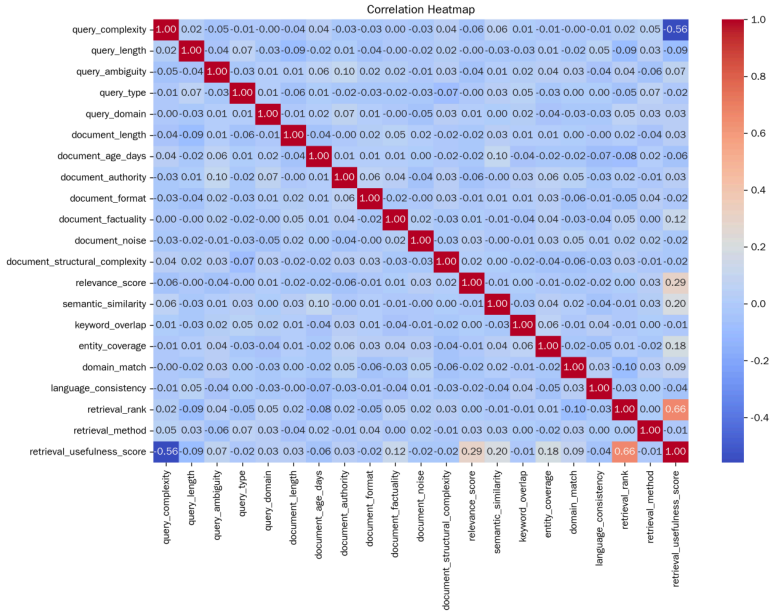

Figure 1. Correlation analysis

DOI: 10.54254/2755-2721/2025.27800

Figure 1 shows the correlation heatmap. From these correlation scores, it can be inferred that there is a strong positive correlation between retroval_rank and retroval_usefilitess_store, indicating that retrieval ranking may have a significant positive impact on retrieval usefulness scores. The higher the retrieval ranking, the higher the retrieval usefulness score may be [5]. And there is a strong negative correlation between query_completeness and retrieval_usefulness_store, indicating that the higher the query complexity, the lower the retrieval usefulness score may be. The correlation between other variables and retrieval_usefilitess_store is relatively weak, and their impact on retrieval usefulness scores is relatively limited. However, they can also reflect to some extent the weak relationship between various factors and retrieval usefulness. Document_factual has a certain positive correlation, indicating that the factual nature of documents may have a positive effect on retrieval usefulness [6].

The Horned Lizard Optimization Algorithm (HLOA) is a metaheuristic optimization algorithm inspired by the survival behavior of horned lizards. Its core principle simulates the efficient hunting and adaptation strategies of horned lizards in desert environments [7]. The network architecture of HLOA is shown in Figure 2. During the algorithm initialization phase, the solution space of the problem to be optimized is mapped to a "habitat", and randomly generated candidate solutions are used as "horned lizard individuals". The search process is divided into two stages: exploration and development. In the exploration stage, the behavior of the horned lizard foraging on a large scale is simulated. By randomly adjusting the step size and direction, a global search is conducted in the solution space to avoid falling into local optima; During the development phase, it simulates the precise sprint of a horned lizard after locking onto its prey, narrowing down the search range based on the current optimal solution and improving local search accuracy through fine adjustments [8]. In addition, the algorithm introduces a "temperature regulation" mechanism to adaptively balance the weights of exploration and development based on the iterative process - initially focusing on global exploration to cover a wider solution space, and later strengthening local development to refine the optimal solution. This strategy of simulating biological adaptive behavior enables HLOA to have strong global optimization ability and convergence speed in complex optimization problems.

3. Method

3.1. HLOA

The Horned Lizard Optimization Algorithm (HLOA) is a metaheuristic optimization algorithm inspired by the survival behavior of horned lizards. Its core principle simulates the efficient hunting and adaptation strategies of horned lizards in desert environments [7]. The network architecture of HLOA is shown in Figure 2.

DOI: 10.54254/2755-2721/2025.27800

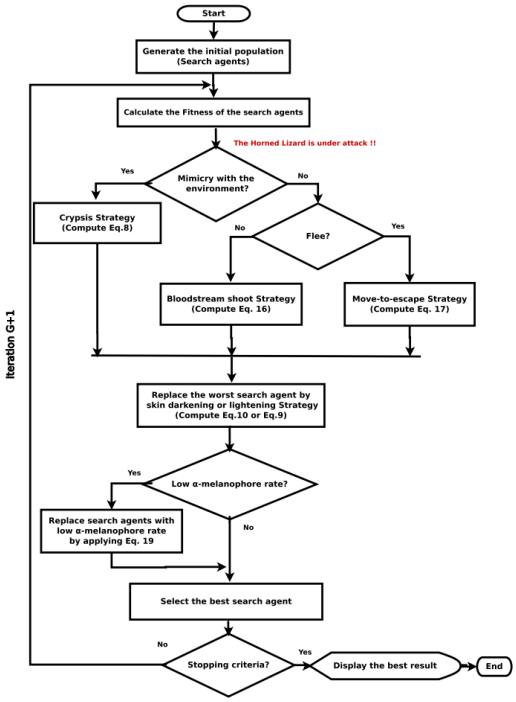

Figure 2. The network architecture of HLOA

During the algorithm initialization phase, the solution space of the problem to be optimized is mapped to a "habitat", and randomly generated candidate solutions are used as "horned lizard individuals". The search process is divided into two stages: exploration and development. In the exploration stage, the behavior of the horned lizard foraging on a large scale is simulated. By randomly adjusting the step size and direction, a global search is conducted in the solution space to avoid falling into local optima; During the development phase, it simulates the precise sprint of a horned lizard after locking onto its prey, narrowing down the search range based on the current optimal solution and improving local search accuracy through fine adjustments [8]. In addition, the algorithm introduces a "temperature regulation" mechanism to adaptively balance the weights of exploration and development based on the iterative process - initially focusing on global exploration to cover a wider solution space, and later strengthening local development to refine the optimal solution.

3.2. CNN

Convolutional Neural Network (CNN) is a deep learning model inspired by the biological visual system. Its core components include convolutional layers, pooling layers, and fully connected layers. The convolutional layer performs local convolution operations on the input data by sliding convolution kernels to extract local features, and the same convolution kernel shares weights throughout the input space, significantly reducing the number of parameters. The pooling layer compresses the feature map dimension through downsampling, preserving key information while enhancing translation invariance [9]. Finally, high-level features are mapped to the output target through fully connected layers.

DOI: 10.54254/2755-2721/2025.27800

3.3. BIGRU

Bidirectional Gated Recurrent Unit (BIGRU) is a bidirectional extension model based on Gated Recurrent Unit (GRU), designed specifically for processing sequential data. Its core inherits the gating mechanism of GRU, dynamically regulating information flow through update gates and reset gates: the update gate determines how much historical information to retain, and the reset gate controls whether to ignore past states, effectively alleviating the gradient vanishing problem of traditional RNNs and enhancing the ability to capture long sequence dependencies. At the same time, BIGRU introduces a bidirectional structure that includes two GRU units, forward and backward. The forward unit processes information in sequential order, while the backward unit processes it in reverse order. The two outputs are concatenated and used as the final feature [10]. This design enables the model to utilize both past and future contextual information to more accurately understand the bidirectional associations of elements in the sequence. The network structure of BIGRU is shown in Figure 3.

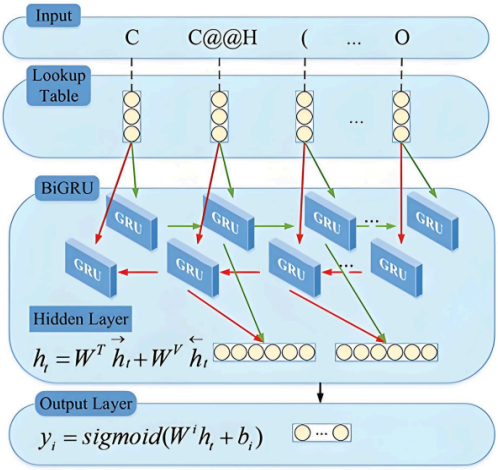

Figure 3. The network structure of BIGRU

3.4. HLOA-CNN-BIGRU

The BIGRU model optimized by HLOA and CNN is a hybrid model that combines metaheuristic optimization and deep learning. In its core architecture, CNN extracts localized features through convolutional kernels; BIGRU models contextual dependencies bidirectionally.

HLOA plays a role in parameter optimization during this process: by simulating the exploration development strategy of the horned lizard, it adaptively adjusts key parameters such as the convolutional kernel size, stride, hidden layer dimension of BIGRU, and gate weights of CNN. The collaboration of the three enables the model to possess local feature capture capability, global dependency modeling capability, and parameter optimization efficiency, making it suitable for prediction tasks of complex sequence data, especially for RAG system retrieval quality evaluation scenarios.

4. Result

In terms of parameter settings for the model, the number of search agents is 8, the maximum number of iterations is 6, the optimization dimension is 3, the learning rate ranges from 1e-4 to 1e-2, the number of BiGRU neurons ranges from 10 to 30, and the L2 regularization parameter ranges from

DOI: 10.54254/2755-2721/2025.27800

1e-4 to 1e-1. The training adopts Adam optimization algorithm, with a maximum training round of 5 in the main program and 500 in the objective function. The initial learning rate is determined by the optimal value obtained from optimization, and the learning rate scheduling method is piecewise. The learning rate decay factor is 0.1, and the decay period is 4 in the main program and 400 in the objective function. The L2 regularization parameter is the optimal value obtained from optimization, and each round of training will shuffle the data. In the network structure, there are two convolutional layers with kernel sizes of [2,1], 16 and 32 filters respectively, and the activation function is relu; The number of neurons in the BiGRU layer is the optimal value obtained through optimization, and the output mode is last; It also includes sequence folding layer, sequence unfolding layer, flattening layer, fully connected layer, and regression layer. In terms of data processing, the training set accounts for 70% of the total data, which is normalized to a range of 0 to 1 using mapminmax.

The comparative models used in this article include decision trees, random forests Adaboost, Gradient Boosting Trees, ExtraTrees, CatBoost, XGBoost, LightGBM, and KNN. The experimental results are shown in Table 2, and the comparison bar chart of the indicators of each model is shown in Figure 4.

Table 2. The results of the comparative experiment

Model	MSE	RMSE	MAE	MAPE	R^2
Decision tree	209.006	14.457	11.384	43.368	0.656
Random Forest	105.254	10.259	8.358	26.692	0.803
Adaboost	79.913	8.939	7.189	27.83	0.861
Gradient Boosting Tree (GBDT)	73.064	8.548	6.923	27.185	0.874
ExtraTrees	85.037	9.222	7.534	27.082	0.84
CatBoost	131.626	11.473	9.182	27.276	0.784
XGBoost	49.006	7	5.703	26.013	0.919
LightGBM	114.129	10.683	8.489	32.406	0.809
KNN	52.506	7.246	5.759	25.826	0.909
HLOA-CNN-BIGRU	28.617	5.349	4.401	17.355	0.952

From various indicators, HLOA-CNN-BIGRU performs better than all other models. On MSE, the value of HLOA-CNN-BIGRU is 28.617, much lower than the 209.006 of decision trees, 105.254 of random forests, 79.913 of Adaboost, 73.064 of gradient boosting trees, 85.037 of ExtraTrees, 131.626 of CatBoost, 114.129 of LightGBM, 49.006 of XGBoost, and 52.506 of KNN. In terms of RMSE, HLOA-CNN-BIGRU is 5.349, which is lower than decision tree's 14.457, random forest's 10.259, Adaboost's 8.939, gradient boosting tree's 8.548, ExtraTrees' 9.222, CatBoost's 11.473, LightGBM's 10.683, XGBoost's 7, and KNN's 7.246. In the MAE index, HLOA-CNN-BIGRU is 4.401, which is lower than decision tree's 11.384, random forest's 8.358, Adaboost's 7.189, gradient boosting tree's 6.923, ExtraTrees' 7.534, CatBoost's 9.182, LightGBM's 8.489, XGBoost's 5.703, and KNN's 5.759. On MAPE, HLOA-CNN-BIGRU is 17.355, which is the lowest among all models, lower than decision tree's 43.368, random forest's 26.692, Adaboost's 27.83, gradient boosting tree's 27.185, ExtraTrees' 27.082, CatBoost's 27.276, LightGBM's 32.406, XGBoost's 26.013, and KNN's 25.826. In the R^2 index, HLOA-CNN-BIGRU reached 0.952, higher than decision tree's 0.656, random forest's 0.803, Adaboost's 0.861, gradient boosting tree's 0.874, ExtraTrees' 0.84, CatBoost's 0.784, LightGBM's 0.809, XGBoost's 0.919, and KNN's 0.909.

DOI: 10.54254/2755-2721/2025.27800

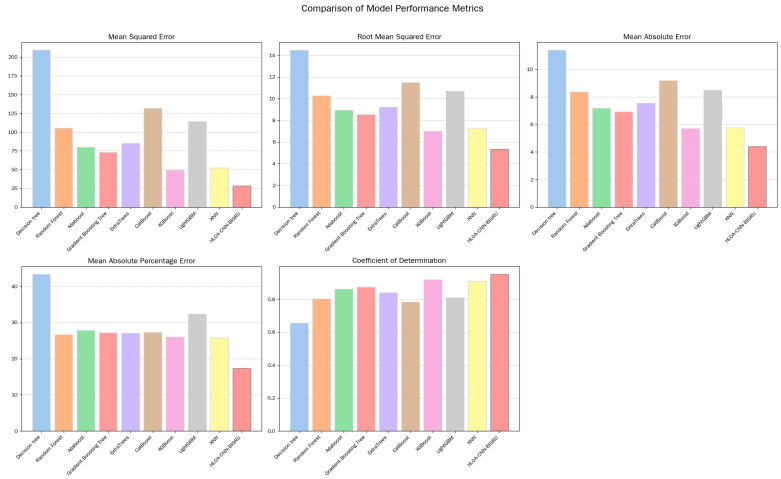

Figure 4. Tthe comparison bar chart of the indicators of each model

To address the issues of query complexity, document noise, and high cost and lag in traditional manual evaluation of retrieval quality in Retrieval Enhanced Generative (RAG) systems, this paper proposes a retrieval quality prediction model that integrates the Lizard Optimization Algorithm (HLOA), Convolutional Neural Network (CNN), and Bidirectional Gated Recurrent Unit (BIGRU). Correlation analysis shows that retrieval rank is strongly positively correlated with retrieval usefulness score, meaning that the higher the retrieval rank, the better the usefulness of the retrieval results; And there is a strong negative correlation between query complexity and retrieval usefulness score, with higher query complexity indicating a decrease in retrieval usefulness. Performance comparison experiments show that the HLOA-CNN-BIGRU model outperforms eight comparison models including decision tree, random forest, and Adaboost in key indicators. Not only is the MSE, RMSE, MAE, MAPE error index significantly lower, but the determination coefficient R^2 reaches 0.952, demonstrating more accurate and stable retrieval quality prediction ability.

This article innovatively integrates the parameter optimization advantages of HLOA with the feature extraction capabilities of CNN and BIGRU, providing a new model architecture for predicting the retrieval quality of RAG systems; Effectively solving the pain point of traditional manual evaluation being unable to meet real-time optimization, it can provide a basis for dynamically adjusting retrieval strategies in RAG systems by accurately predicting retrieval quality, thereby improving the accuracy and reliability of generated content in scenarios such as intelligent Q&A and knowledge assistants, and promoting the more efficient implementation of RAG technology in practical applications.

5. Conclusion

This article applies the improved least squares support vector machine (SBO-LSSVM) algorithm of state optimization algorithm to parameter optimization and quality prediction of metal selective laser melting (SLM) process. The study first identified the key influencing factors through indicator correlation analysis. From the correlation heatmap, it can be observed that the correlation color between laser power, scanning speed, and layer thickness with the relative density of the part is darker. This directly proves the key regulatory role of these three parameters on the relative density of the part in SLM process, providing clear direction for subsequent parameter optimization. To verify the performance of SBO-LSSVM, the experiment used decision tree, random forest, XGBoost, and LightGBM as control models. The results showed that SBO-LSSVM performed the

DOI: 10.54254/2755-2721/2025.27800

best in all evaluation indicators: its MSE, RMSE, MAE, and MAPE were 1.089, 1.043, 0.766, and 0.867, respectively, which were the minimum values among all models. Compared with decision tree (1.952, 1.397, 1.031, 1.159), random forest (2.195, 1.482, 1.002, 1.13), and LightGBM (2.017, 1.42, 1.018, 1.149), SBO-LSSVM significantly reduced prediction errors, especially MSE, which was about 24.6% lower than the suboptimal XGBoost (1.445).

In terms of R^2 index of data fitting ability, SBO-LSSVM's 0.855 is also better than all control models. The R^2 of decision tree, random forest, XGBoost, and LightGBM are 0.786, 0.751, 0.8, and 0.764, respectively, which fully demonstrates its stronger ability to capture the actual data patterns of SLM process. Even though XGBoost performs relatively well on MAE (0.819) and R^2 (0.8), it still falls short of SBO-LSSVM; And the random forest showed the weakness of the highest MSE and the lowest R^2, further confirming the comprehensive advantages of SBO-LSSVM in prediction accuracy and fitting degree. This study not only provides more accurate quality prediction and parameter optimization tools for SLM technology, effectively reducing production losses and quality defects caused by unreasonable parameters, but also provides a feasible path for the optimization and upgrading of machine learning models in the field of additive manufacturing, which has important practical value for promoting the industrial application of SLM technology to high precision.

References

[1] Huly, Oz, David Carmel, and Oren Kurland. "Predicting RAG Performance for Text Completion." Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2025.
[2] Zhao, Shengming, et al. "Towards understanding retrieval accuracy and prompt quality in rag systems." arXiv preprint arXiv: 2411.19463 (2024).
[3] Veturi, Sriram, et al. "Rag based question-answering for contextual response prediction system." arXiv preprint arXiv: 2409.03708 (2024).
[4] Chan, Chi-Min, et al. "Rq-rag: Learning to refine queries for retrieval augmented generation." arXiv preprint arXiv: 2404.00610 (2024).
[5] Shi, Yunxiao, et al. "Enhancing retrieval and managing retrieval: A four-module synergy for improved quality and efficiency in rag systems." arXiv preprint arXiv: 2407.10670 (2024).
[6] He, Jacky, et al. "Context-Guided Dynamic Retrieval for Improving Generation Quality in RAG Models." arXiv preprint arXiv: 2504.19436 (2025).
[7] Jiang, Ziyan, Xueguang Ma, and Wenhu Chen. "Longrag: Enhancing retrieval-augmented generation with long-context llms." arXiv preprint arXiv: 2406.15319 (2024).
[8] Ampazis, Nicholas. "Improving RAG quality for large language models with topic-enhanced reranking." IFIP international conference on artificial intelligence applications and innovations. Cham: Springer Nature Switzerland, 2024.
[9] Yang, Xiao, et al. "Crag-comprehensive rag benchmark." Advances in Neural Information Processing Systems 37 (2024): 10470-10490.
[10] Zhang, Zihan, Meng Fang, and Ling Chen. "Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering." arXiv preprint arXiv: 2402.16457 (2024).