Proceedings of The 6th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/2026.30304

Prediction of RAG System Retrieval Strategy Selection Based on Multihead-Attention Optimization of LSTM Algorithm

Qi He

Department of Industrial Engineering, Tsinghua University, Beijing, China
moziofmoon@163.com

Abstract. Against the backdrop of the increasing popularity of large language models in knowledge-intensive tasks, Retrieval Enhancement generation (RAG) technology has become a key technical path to break through model illusions and enhance the credibility of output. However, when dealing with the RAG retrieval strategy prediction task, existing machine learning algorithms generally have problems such as unbalanced feature weight configuration and insufficient capture of long sequence dependencies. To this end, this paper proposes an LSTM classification algorithm based on Multihead-Attention optimization. Firstly, data preprocessing is completed through correlation analysis and violin graph analysis, and then performance comparison experiments are conducted with multiple machine learning algorithms. The results show that the proposed Multihead-Attention-LSTM model performs the best in all evaluation indicators: The accuracy rate reached 0.853, the recall rate and precision rate were 0.853 and 0.854 respectively, the F1 value was 0.853, and the AUC value was 0.97. It comprehensively outperformed integrated models such as ExtraTrees and XGBoost, as well as traditional models like Random Forest, GBDT, and decision tree, and was significantly superior to Naive Bayes and KNN. This model, by leveraging the advantages of the multi-head attention mechanism and LSTM, demonstrates outstanding superiority in classification performance and generalization ability. It effectively verifies its applicability in the classification task of retrieval strategies and provides efficient and feasible algorithmic support for the intelligent optimization of retrieval strategies in the RAG system. It has significant practical value for enhancing the output reliability of large language models in knowledge-intensive tasks.

Keywords: RAG, Machine learning algorithm, RAG retrieval strategy, Multihead-Attention

1. Introduction

With the wide application of large language models in knowledge-intensive tasks, retrieval enhancement generation technology has become the core solution to address model illusions and enhance the credibility of output [1]. At present, the retrieval link of RAG systems mostly adopts a single fixed strategy, such as pure vector retrieval or keyword retrieval. However, in actual application scenarios, there are significant differences in query characteristics, document library attributes and system states: Simple and general long queries are more suitable for the semantic matching ability of vector retrieval. Short queries in professional fields rely on the precise

DOI: 10.54254/2755-2721/2026.30304

positioning of keyword retrieval. Large-scale document libraries require hierarchical retrieval to reduce computational costs, while complex multi-intent queries often need a hybrid strategy to balance recall rate and accuracy [2]. Fixed strategies are difficult to cover the demands of diverse scenarios, resulting in fluctuations in retrieval efficiency and unstable generation quality. Therefore, dynamically selecting the optimal retrieval strategy based on real-time scenarios has become a key direction for the performance optimization of RAG systems and is also a core issue that urgently needs to be addressed in the current technology implementation process [3].

Machine learning algorithms provide an effective solution for the dynamic selection of retrieval strategies in RAG systems [4]. Traditional rule-based strategy selection relies on manual preset logic and is unable to handle the nonlinear correlations of multiple variables in complex scenarios. In contrast, machine learning algorithms can automatically learn the mapping relationship between multi-dimensional features such as query complexity, document library size, and historical retrieval accuracy and the optimal strategy through a data-driven approach [5]. For instance, the decision tree algorithm can visually present the influence weights of features on strategy selection, the support vector can effectively handle classification tasks in high-dimensional feature Spaces, and the Naive Bayes algorithm is suitable for rapid prediction in small sample scenarios. These algorithms, through training on historical interaction data, can achieve strategy prediction in real-time scenarios, significantly reducing the cost of manual intervention, while enhancing the adaptability and accuracy of strategy selection, providing technical support for the intelligent upgrade of the retrieval link in the RAG system [6].

When dealing with the RAG retrieval strategy prediction task, existing machine learning algorithms still have problems such as unreasonable feature weight distribution and insufficient capture of long sequence dependencies: traditional algorithms mostly regard each feature as an independent variable, ignore the association between query characteristics and document library attributes, and it is difficult to effectively model the dynamic dependency relationship between features. To this end, this paper proposes an LSTM classification algorithm based on Multihead-Attention optimization. This algorithm utilizes the long short-term memory capability of the LSTM network to capture the temporal correlations and long-term dependencies in the feature sequence. Meanwhile, through the Multihead-Attention mechanism, The importance of different feature dimensions is dynamically weighted to further enhance the accuracy and robustness of the retrieval strategy selection in the RAG system, providing a new algorithmic path for the efficient implementation of RAG technology in diverse practical scenarios.

2. Data sources

This dataset contains a total of 723 samples and can be used for the optimization of retrieval strategy selection in the RAG system. The data covers seven characteristic variables, namely query complexity, query length, domain specificity, document library size, average document length, document update frequency and historical retrieval accuracy. The predictor variable is the optimal retrieval strategy, which includes four types: vector retrieval, keyword retrieval, mixed retrieval and hierarchical retrieval. It can be used for the training and validation of classification models related to the selection of retrieval strategies in the RAG system. Some datasets are selected for display, as shown in Table 1.

DOI: 10.54254/2755-2721/2026.30304

Table 1. A partial dataset

query_complexity	query_length	domain_specificity	doc_library_size	avg_doc_length	update_frequency	historical_accuracy	optimal_strategy
7	1	0.94	36973	1544	0.214	48.4	Keyword search
8.3	1	0.61	9682	4192	0.29	45.8	Hybrid retrieval
10	17	0.38	469413	8235	0.063	79.3	Hierarchical search
3.3	25	0.17	853302	8302	0	77.8	Vector retrieval
9.4	7	1	151965	6954	0.123	56.3	Hybrid retrieval

Output the violin diagrams of each variable, as shown in Figure 1.

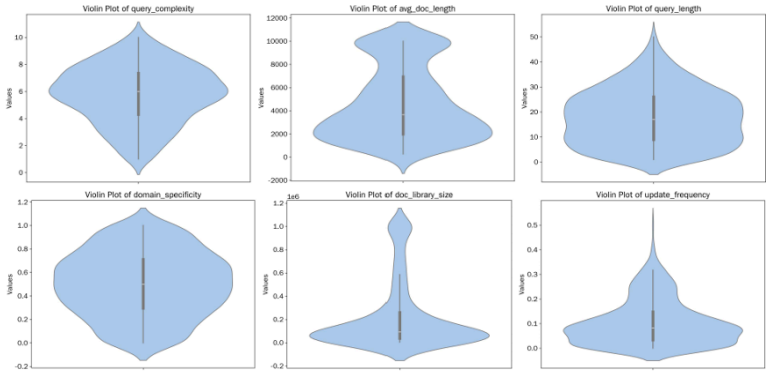

Figure 1. The violin diagrams of each variable

Correlation analysis was conducted on each variable, and a correlation heat map was drawn, as shown in Figure 2. It can be known from the correlation coefficients that the correlation coefficient between the optimal strategy and the query complexity is 0.35, the correlation coefficient with the query length is 0.21, the correlation coefficient with the document library size is 0.22, and the correlation coefficient with the average document length is 0.30. These positive correlations indicate that as the query complexity, query length, document library size, and average document length increase, The trend of the emergence of the optimal strategy may also rise. However, its correlation coefficient with domain professionalism is -0.15, showing a relatively weak negative correlation. The correlation coefficient with update frequency is -0.40 and the correlation coefficient with historical accuracy rate is -0.43, showing a strong negative correlation.

DOI: 10.54254/2755-2721/2026.30304

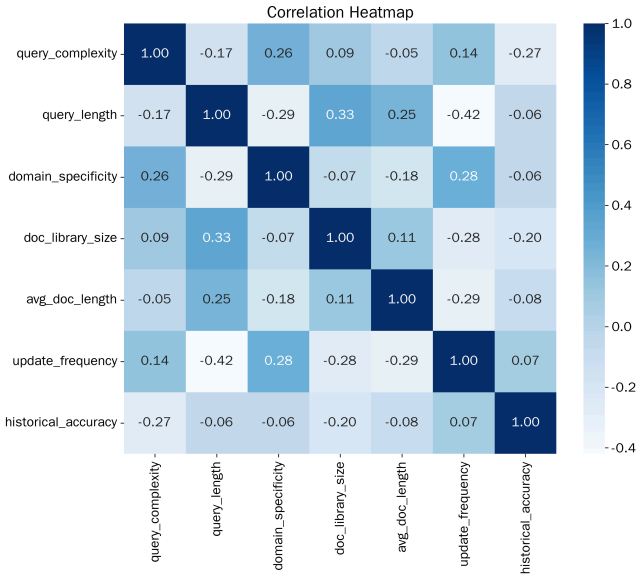

Figure 2. The correlation heat map

3. Method

3.1. Multihead-Attention

Multihead-Attention is the core component of the Transformer model. Its core lies in capturing the global dependency of the input data through a multi-head parallel self-attention mechanism [7]. The core steps revolve around query, key, and value: first, the input embedding vector is linearly transformed to generate three matrices of Q, K, and V, and then the three are split by the number of heads. For each group, the similarity is calculated by scaling the dot product attention, the attention weight is obtained through Softmax, and then the single-head output is obtained by weighted summation. Finally, all the outputs of the heads are concatenated and integrated into the final result through linear transformation [8]. The multi-head design enables the model to simultaneously focus on feature associations of different dimensions and distances. The network structure of Multihead-Attention is shown in Figure 3.

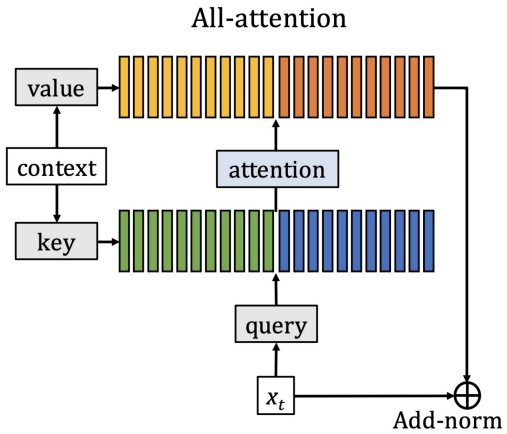

Figure 3. The network structure of Multihead-Attention

DOI: 10.54254/2755-2721/2026.30304

3.2. LSTM

LSTM is a time series model designed to solve the vanishing/exploding gradient problem of traditional RNN. Its core lies in precisely controlling the storage and flow of information through a gating mechanism. Its core structure includes cell states and three gates: The forgetting gate determines whether to discard historical information in the cell state through Sigmoid activation. The input gate first filters new information through Sigmoid, then generates candidate information through tanh and updates the cell state. The output gate filters information in the cell state through Sigmoid and tanh and generates the current moment output [9]. The linear transmission characteristics of cell states enable the stable transmission of long sequence information, while the gating mechanism flexibly controls the increase or decrease of information, retaining key long-term dependencies while filtering out redundant short-term noise, thus becoming a classic model for processing time series data. The network structure of LSTM is shown in Figure 4.

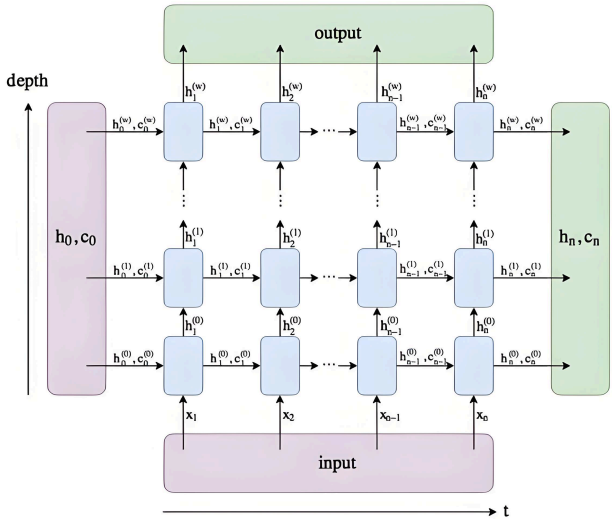

Figure 4. The network structure of LSTM

3.3. Multihead-Attention-LSTM

The Multihead-Attention-LSTM classification algorithm is a hybrid architecture that integrates the advantages of both models. Its core lies in using Attention to capture global dependencies and LSTM to capture temporal dependencies, thereby enhancing the feature representation ability of classification tasks [10]. The typical process is as follows: First, the input time series data is input into the Multihead-Attention layer, and the global associated features are extracted through the multi-head mechanism to make up for the deficiency of the global capture ability of LSTM; Then, input the global feature sequence output by Attention into the LSTM layer, and use the gating mechanism to model the temporal evolution law of the features while retaining the sequential information of the sequence. Finally, perform linear transformation and Softmax activation on the output of LSTM to obtain the classification probability distribution.

4. Result

In terms of parameter Settings, the training model adopts the adam gradient descent algorithm, with the maximum number of iterations set to 150, the batch size to 128, and the initial learning rate to

DOI: 10.54254/2755-2721/2026.30304

0.001. The learning rate is scheduled in segments, and the descent factor is 0.1. After 1200 training sessions, the learning rate is updated to the initial value multiplied by the descent factor. When dividing the dataset, the proportion of the training set is 0.7. Whether to shuffle the dataset can be controlled through annotations. When the flag bit flag_conusion is set to 1, the confusion matrix will be displayed. In the model structure, the output dimension of the LSTM layer is 10 and the output mode is last.

Decision tree, Random Forest, AdaBoost, Naive Bayes, KNN, ExtraTrees, XGBoost and GBDT were used as comparison models, and the model results are shown in Table 2. The comparison results of each indicator are shown in Figure 5.

Table 2. The results of the comparative experiment

Model	Accuracy	Recall	Precision	F1	AUC
Decision tree	0.76	0.76	0.76	0.759	0.878
Random Forest	0.806	0.806	0.809	0.807	0.967
AdaBoost	0.719	0.719	0.74	0.715	0.902
Naive Bayes	0.691	0.691	0.706	0.688	0.898
KNN	0.516	0.516	0.491	0.49	0.753
ExtraTrees	0.816	0.816	0.815	0.815	0.964
XGBoost	0.82	0.82	0.825	0.821	0.963
GBDT	0.793	0.793	0.792	0.792	0.936
Multihead-Attention-LSTM(Our model)	0.853	0.853	0.854	0.853	0.97

It can be known from the comparison results that the Multihead-Attention-LSTM proposed in this paper performs the best in all evaluation indicators and comprehensively outperforms traditional machine learning models such as decision trees and random forests, as well as ensemble learning models. In terms of Accuracy, this model achieved 0.853, which is higher than ExtraTrees' 0.816 and XGBoost's 0.82, and also significantly outperforms Random Forest's 0.806 and GBDT's 0.793. It far exceeds the 0.691 of Naive Bayes and 0.516 of KNN. In terms of Recall and Precision dimensions, this model is 0.853 and 0.854 respectively. It not only outperforms models such as ExtraTrees and XGBoost in the ensemble class, but also significantly leads models like decision trees and AdaBoost. KNN performed the worst in these two indicators, only 0.516 and 0.491. In terms of F1 values, this model ranks first with 0.853, significantly outperforming XGBoost's 0.821, ExtraTrees' 0.815, and Random Forest's 0.807. The F1 values of other models are all below 0.8, with the lowest KNN being only 0.49. In terms of the AUC metric, this model reached 0.97, slightly higher than the 0.967 of Random Forest and the 0.964 of ExtraTrees, and higher than the 0.963 of XGBoost. At the same time, it far exceeded GBDT, AdaBoost, Naive Bayes and KNN. Among them, the AUC of KNN is only 0.753. Overall, the Multihead-Attention-LSTM proposed in this paper, by combining the multi-head attention mechanism with LSTM, demonstrates significant advantages in classification accuracy, recall rate, precision rate, comprehensive evaluation indicators, and generalization ability, fully verifying its effectiveness in the classification task of retrieval strategies.

DOI: 10.54254/2755-2721/2026.30304

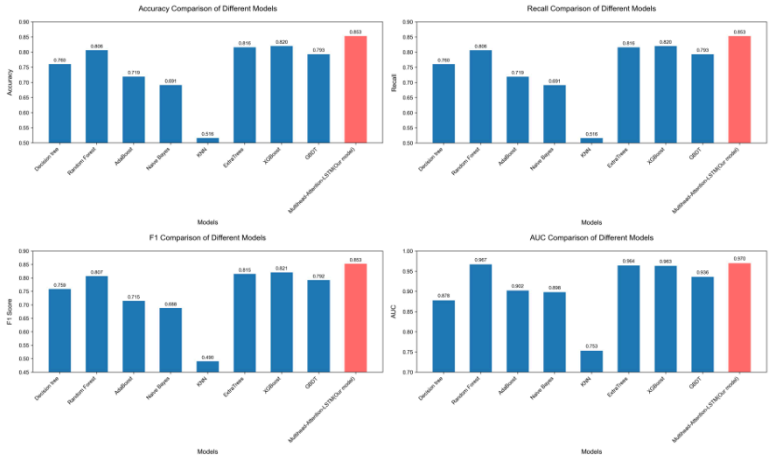

Figure 5. The comparison results of each indicator

5. Conclusion

In the context of large language models deeply empowering knowledge-intensive tasks, Retrieval enhancement generation (RAG) technology has become a key path to break through model illusions and enhance the credibility of output. However, when dealing with the prediction tasks of RAG retrieval strategies, the current mainstream machine learning algorithms still face problems such as unbalanced feature weight configuration and insufficient capture of long sequence dependencies. To this end, this paper proposes an LSTM classification algorithm integrating Multihead-Attention, lays the research foundation through correlation analysis and violin graph analysis, and conducts performance comparisons with multiple machine learning models. The experimental results show that the proposed Multihead-Attention-LSTM model performs outstandingly in all evaluation indicators: The Accuracy reached 0.853, which was superior to integrated models such as ExtraTrees (0.816) and XGBoost (0.82), and far exceeded Naive Bayes (0.691) and KNN (0.516). The Recall and Precision were 0.853 and 0.854 respectively, significantly leading models such as decision tree and AdaBoost, while KNN ranked at the bottom with 0.516 and 0.491. With an F1 value of 0.853, it ranks first, and an AUC of 0.97, both comprehensively surpassing traditional machine learning and ensemble learning models. This model, by integrating the advantages of the multi-head attention mechanism and LSTM, has constructed significant advantages in classification accuracy, recall rate, precision rate, comprehensive evaluation and generalization ability, fully verifying its effectiveness in the RAG retrieval strategy classification task and providing an efficient technical solution for the precise adaptation of the RAG system retrieval strategy. It also provides a solid technical support for the continuous improvement of model credibility in knowledge-intensive tasks.

References

[1] Salemi, Alireza, and Hamed Zamani. "Evaluating retrieval quality in retrieval-augmented generation." Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2024.

[2] Leng, Quinn, et al. "Long context rag performance of large language models." arXiv preprint arXiv: 2411.03538 (2024).

[3] 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).

[4] Zhao, Shengming, et al. "Towards understanding retrieval accuracy and prompt quality in rag systems." arXiv preprint arXiv: 2411.19463 (2024).

DOI: 10.54254/2755-2721/2026.30304

[5] Jin, Bowen, et al. "Long-context llms meet rag: Overcoming challenges for long inputs in rag." arXiv preprint arXiv: 2410.05983 (2024).
[6] Mengmeng, Su, et al. "An Effective Retrieval Method to Improve RAG Performance." 2024 7th International Conference on Data Science and Information Technology (DSIT). IEEE, 2024.
[7] Tang, Yixuan, and Yi Yang. "Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries." arXiv preprint arXiv: 2401.15391 (2024).
[8] Jiang, Wenqi, et al. "Rago: Systematic performance optimization for retrieval-augmented generation serving." Proceedings of the 52nd Annual International Symposium on Computer Architecture. 2025.
[9] Oche, Agada Joseph, et al. "A systematic review of key retrieval-augmented generation (rag) systems: Progress, gaps, and future directions." arXiv preprint arXiv: 2507.18910 (2025).
[10] Luo, Kun, et al. "Does RAG Really Perform Bad For Long-Context Processing?." arXiv preprint arXiv: 2502.11444 (2025).