Keyword-Enhanced Semantic Retrieval and Multi-Dimensional Relevance Ranking in RAG

Kui Zhao*
Shenyang Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Liaoning Province Human-Computer Interaction System Engineering Research Center Based on Digital Twin
Shenyang, China
zhaokui@sict.ac.cn
*Corresponding author

Guoyu Wang
Shenyang Institute of Computing Technology, Chinese Academy of Sciences
University of Chinese Academy of Sciences
Shenyang, China
wangguoyu221@mails.ucas.ac.cn

2024 10th International Conference on Computer and Communications (ICCC) | 979-8-3315-0707-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCC62609.2024.10941711

***Abstract*—In the task of generating medical hot topics, efficiently and accurately retrieving relevant information based on hot keywords and generating credible topics is a challenging problem. Addressing the limitations of existing retrieval and ranking methods in terms of accuracy and relevance, this paper proposes a RAG-based keyword-enhanced semantic retrieval and multi-dimensional relevance ranking approach. First, we construct keyword vectors using the BGE model and perform fast similarity searches to achieve preliminary retrieval. Then, we optimize the relevance of the retrieval results by setting a similarity threshold and applying a cluster and de-duplication strategy. Building on this, multi-dimensional similarity measurement is introduced, combining a cross-encoder model and feature extraction to implement reordering, further improving the accuracy of retrieval results. Experimental results show that this method significantly enhances RAG's performance in generating hot topics from medical news and information, providing an effective solution for complex information retrieval tasks.**

Keywords—Medical Information Retrieval; Hot Topic Generation; Keyword-Enhanced; Semantic Retrieval; Multi-Dimensional Reordering; RAG

I. INTRODUCTION

In the information society, the timeliness and accuracy of medical news information become particularly important. The automated processing and interpretation of medical news information through the use of LLM, RAG, and other technologies can help medical practitioners, researchers, policymakers, and the general public keep abreast of the latest developments in the medical field[1]. Medical information plays a key role in disease prevention, health education, and public health response. At the same time, the generation and dissemination of hot topics also occupy an important position in modern news, especially in the rapid response to public health emergencies and emerging medical advances. The accurate generation of hot topics helps to enhance the public's attention and cognition of health issues. Therefore, how to quickly generate reliable and timely hot topics based on a large number of medical information data has become one of the important directions of current research.

At present, LLMS, such as ChatGPT, Llama[2], Qwen[3], Gemini [4], Gemma[5], etc., have good natural language understanding ability, and at the same time, in the news Generation system based on RAG (Retrieval-Augmented Generation) technology[6], The accuracy of retrieval and sorting algorithms directly affects the quality of generated topics. Most of the existing retrieval and ranking methods adopt traditional information retrieval techniques, which usually rely on keyword matching or basic vector similarity calculation, which will face two main problems in large-scale data environment: first, the retrieval correlation is not high, and it is impossible to accurately find the content that best matches the hot keywords; Second, the lack of a comprehensive assessment of the similarity of different dimensions makes it difficult to achieve the best balance in terms of content relevance, diversity and domain specificity[7]. The limitation of the existing technology makes the quality and credibility of the generated hot topics insufficient. Especially in the complex medical field, it is difficult for traditional methods to meet the accuracy and timeliness requirements of generating hot topics. Therefore, it is necessary to improve the existing retrieval and reordering methods to improve the relevance and accuracy of generating hot topics.

This study aims to propose an optimized retrieval and reordering method based on RAG technology to improve the quality and accuracy of hot topic generation. Firstly, according to the characteristics of medical news, we designed a Keyword-Enhanced Semantic Retrieval(KESR), which improved the accuracy and response speed of retrieval by constructing keyword vectors, batch retrieval, and similarity threshold setting. Second, we develop a reordering algorithm based on a Multi-Dimensional Relevance Ranking (MDRR) that utilizes a high-level semantic model (e.g. encoder BERT) combined with domain-specific features (e.g. medical term match, text source authority, etc.) to achieve an effective balance between the relevance, accuracy, and diversity of the generated results.

II. RELATED WORK

A. Existing retrieval technology

Traditional reordering methods are mainly based on statistical models or hand-designed features, such as BM25, TF-IDF, etc.

With the development of natural language processing and deep learning, vector retrieval has gradually replaced the dominant position of keyword retrieval in complex retrieval tasks. Vector retrieval transforms text, terms, or queries into high-dimensional vectors and uses the distance or similarity between the vectors (such as cosine similarity, Euclidean distance, etc.) to determine correlation.

In RAG, knowledge retrieval usually adopts a variety of retrieval strategies to enhance the depth of retrieval and improve the accuracy of retrieval results. Hybrid search[8] usually rewrites the user query question to generate one or more queries. Common search methods include database query, vector search, QA search, knowledge graph search, plug-in search, keyword search, and so on[9]. After searching by multiple retrieval methods, each retrieval method will output TopK search results.

In the large-scale data environment, vector retrieval is faced with the problem of excessive computational overhead, so it is necessary to optimize the index and efficient retrieval algorithms, such as the FAISS vector database[10], for efficient similarity search and clustering of dense vectors.

B. Research progress of reordering algorithm

Re-ranking technology plays an important role in the whole process of Retrieval Augmented Generation (RAG) [11]. In the most primitive RAG approach, a large number of contexts may be retrieved, but not all of them are relevant to the problem. Re-ranking techniques improve the accuracy of RAG systems by reordering and filtering documents, eliminating irrelevant or unimportant documents, and putting relevant documents first.

At present, there are two main reordering methods used in RAG technology, using reordering model and using large Language model (LLM) for re-ranking.

- Among the reordering models, one is Cohere's online model, which is accessible through the API. There are also open-source models such as bge-reranker-base[12] and bge-reanker-large[13]. Unlike embedded models, reordering models take query and context as inputs and directly output similarity scores. It is important to note that the reordering model is optimized using cross-entropy losses, allowing the correlation score not to be limited to a specific range and may even be negative.
- The idea of RankGPT is to perform zero-shot paragraph reordering using LLMS such as ChatGPT or GPT-4 or other LLMS, which applies the permutation generation method and sliding window strategy to effectively reorder paragraphs[14].

In addition, to improve the diversity of results, algorithms such as MMR (Maximal Marginal Relevance) are widely used in sorting post-processing. Deep learning methods have significant advantages in capturing complex semantic relationships and are suitable for scenarios that require high semantic understanding.

C. RAG Technology Overview

1) How RAG works

RAG (Retrieval-Augmented Generation) is a hybrid method that combines retrieval and generation. The core idea is to introduce external knowledge sources in the process of generating answers, by first retrieving relevant content, and then generating answers according to the retrieved context. Specifically, RAG first uses the retrieval module to find the content relevant to the user's query, input the content as context into the generation model, and then generate the answer that meets the query requirements according to the context. This method combines the advantages of retrieval and generation and is able to show high accuracy in tasks where knowledge is incomplete or requires external knowledge support.

2) The application of RAG in generating hot topics

In the hot topic generation scenario, RAG introduces a retrieval module to make the generation model use external medical news information as a reference, so as to improve the accuracy and relevance of the generated topic. Compared with the pure generative model, RAG can dynamically introduce the latest information in the generation process of hot topics, and efficiently generate topics related to medical hot topics in a large-scale data environment. However, the performance of RAG depends on the quality of the search results, so it is necessary to optimize the retrieval and reordering algorithms to further improve the application effect of RAG in the generation of medical information hot topics.

III. METHODS

This thesis is dedicated to addressing the issue of how to effectively employ the vectors of hot words for conducting relevant information retrieval in the vector database, with the aim of extracting the original texts of relevant medical news and information. During the process of generating topics with large models, based on the RAG technology, the collected medical news and information are stored in the vector database. The vectors of keywords are matched with those in the database to retrieve the relevant article contents, thereby providing references for the large model and enhancing the accuracy of topic generation. Hence, the key algorithmic difficulty in this scenario lies in how to precisely match each keyword vector with the paragraph vectors in the database for the optimization of retrieval and re-ranking algorithms.

A. Keyword-Enhanced Semantic Retrieval (KESR)

In this study, a keyword-enhanced semantic retrieval method was proposed to enhance the relevance and accuracy of RAG in retrieving medical news information. The core of this method lies in effectively constructing keyword vectors and performing retrieval through fast similarity search. Then, a similarity threshold is set and the results are aggregated and deduplicated. Figure 1 KESR structure diagram. Here are the specific steps of the KESR method:

1) Keyword vector construction

Vector representation of keywords is the basis of retrieval algorithms. First, we perform word segmentation processing on the collected hot keywords to ensure that each word can generate a vector independently. Using the BGE model, we can effectively capture the semantic features of keywords and generate high-quality vector representations. Specifically, for each $\mathrm{keyword}_k$, we generate its vector representation using the following formula.

$$\mathbf{v}_k = \text{BGE}(\text{keyword}_k) \quad (1)$$

Where $\mathbf{v}_k$ represents the vector of the keyword_k. This process ensures the richness and relevance of information and lays a solid foundation for subsequent retrieval.

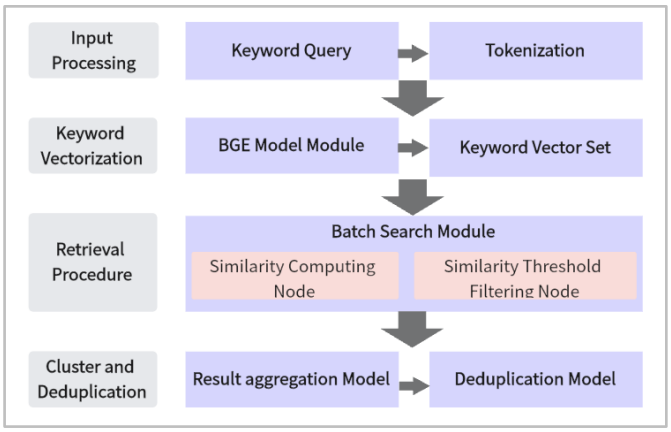

Fig. 1. KESR Structure Diagram

2) Retrieval procedure

In the retrieval stage, we package multiple keyword query vectors into a batch and utilize FAISS's batch query capability to improve retrieval efficiency and reduce the number of interactions with the database. During the retrieval process, we focus on the similarity of vectors and use the cosine similarity as the similarity measure to select the top N text segments closest to the query vector. It is suggested that the chunk-size setting for document knowledge construction be smaller than usual in this scenario, as shorter chunks can improve the accuracy of retrieval. Specifically, for the query vector q and candidate text vector $\mathbf{v}_t$, we calculate the cosine similarity using the following formula:

$$\text{Cosine Similarity}(\mathbf{q}, \mathbf{v}_t) = \frac{\mathbf{q} \cdot \mathbf{v}_t}{\|\mathbf{q}\| \|\mathbf{v}_t\|} \quad (2)$$

After the similarity score is obtained, we set a similarity threshold of θ to ensure that paragraphs of text that are closer to the query vector are filtered out. Only candidate texts that meet the following criteria will be retained:

$$\text{Cosine Similarity}(\mathbf{q}, \mathbf{v}_t) \geq \theta \quad (3)$$

By setting an appropriate threshold, redundant results can be effectively reduced, thus improving the accuracy of retrieval.

3) Cluster and Deduplication

To avoid returning to the same text paragraph repeatedly, we cluster and de-duplicate the search results. We use hash tables or aggregate data structures to quickly de-duplicate and ensure that the aggregated result set is a unique text segment. Specifically, we can express it as:

$$\text{Unique Results} = \text{Unique}(\{\text{result}_1, \text{result}_2, \ldots, \text{result}_N\}) \quad (4)$$

After reprocessing, the text segment with the highest similarity to the keyword is returned preferentially. Overall, our search algorithm optimization process aims to improve the efficiency of the search and the relevance of the results and to provide high-quality candidate text for subsequent reordering steps.

B. Multi-Dimensional Relevance Ranking (MDRR)

In this study, to improve the relevance and accuracy of the search results, we designed a reordering algorithm based on multidimensional similarity measurement. By introducing more advanced models and a variety of evaluation indicators, the algorithm significantly improves the quality and credibility of large model-generating hot topics. Figure 2 MDRR structure diagram. The reordering process is divided into the following steps:

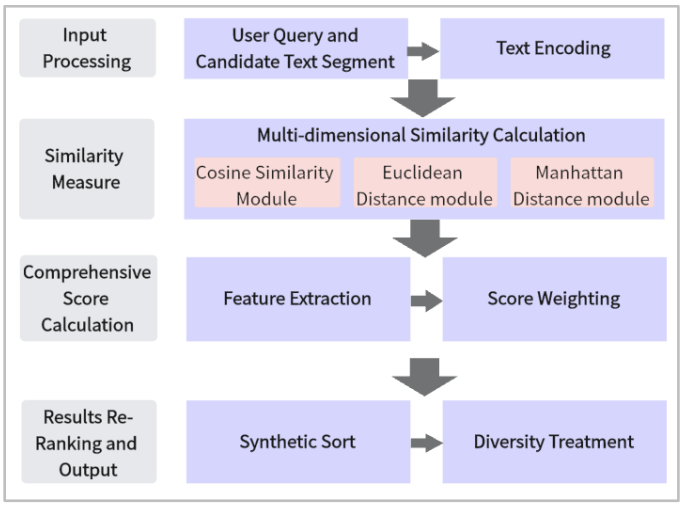

Fig. 2. MDRR Structure Diagram

1) Text encoding and feature extraction

We use a fine-tuned encoder model (e.g. BERT、RoBERTa) to encode query text q and candidate text segment t to capture the subtle semantic relationship between them and output a more accurate correlation score. The specific formula is:

$$\text{Score}_{\text{model}}(q, t) = f_{\text{encoder}}(q, t) \quad (5)$$

In addition, for each candidate text, we also extract other features, such as the matching degree of medical terms and the authority of text sources, to form a feature vector F:

$$\mathbf{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix} \quad (6)$$

2) Multidimensional evaluation and introduction of domain-specific features

In evaluating the similarity of candidate texts, we combine several similarity measures, including cosine similarity, Euclidean distance, and Manhattan distance. The specific calculation formula is as follows:

- Cosine similarity

$$\text{Cosine Similarity}(q, t) = \frac{q \cdot t}{\|q\| \|t\|} \quad (7)$$

- Euclidean distance

$$\text{Euclidean Distance}(q,t) = \sqrt{\sum_{i=1}^{n} (q_i - t_i)^2} \quad (8)$$

- Manhattan distance

$$\text{Manhattan Distance}(q,t) = \sum_{i=1}^{n} |q_i - t_i| \quad (9)$$

We also consider domain-specific characteristics such as authoritativeness of the source of the text, publication time, etc. $d_1, d_2, \ldots, d_m$ is used as an auxiliary scoring index to form a feature vector D:

$$\mathbf{D} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_m \end{bmatrix} \quad (10)$$

3) Comprehensive score calculation

In the comprehensive score calculation stage, we combined the model score, similarity measure and feature extraction results to calculate the final score for each candidate text t. The score is calculated as follows:

$$\text{Score}_{\text{final}}(t) = w_1 \cdot \text{Score}_{\text{model}}(q,t) + w_2 \cdot \text{Cosine Similarity}(q,t) + w_3 \cdot \text{Euclidean Distance}(q,t) + w_4 \cdot \text{Manhattan Distance}(q,t) + \sum_{j=1}^{m} w_j' \cdot d_j \quad (11)$$

Where w_i and w_j' are the weights for each scoring dimension and domain-specific features. By setting these weights reasonably, we ensure the rationality and validity of the overall score.

4) Result in reordering

In the process of reordering the results, we sort the candidate texts according to the comprehensive score, which is specifically expressed as:

$$\text{Ranked Results} = \text{Sort}(\{\text{Score}_{\text{final}}(t_1), \text{Score}_{\text{final}}(t_2), \ldots, \text{Score}_{\text{final}}(t_n)\}) \quad (12)$$

To ensure the diversity of results, we also apply the maximum mutual information reordering (MMR) algorithm, the formula is:

$$\text{MMR}(q,t_i) = \alpha \cdot \text{Score}_{\text{final}}(t_i) - (1-\alpha) \cdot \max_{t_j \in \text{selected}} \text{Similarity}(t_i, t_j) \quad (13)$$

In the above formula, α is the equilibrium parameter and t_j is the selected text. Finally, we will return a finely retrieved and reordered text paragraph to provide a reference for the large model to generate hot topics.

IV. Experimental Results

In this experiment, we use the Retrieval Augmented Generation Assessment (RAGAs) evaluation framework proposed by S. Es et al in September 2023 to evaluate the performance of RAG systems. The innovation of RAGAs is that it does not rely on manually annotated standard answers, but makes use of the underlying Large Language model (LLM) for automatic evaluation.

A. Dataset

Using Hugging Face public data sets, Datasets: exploding_gradients/amnesty_qa. The data set used contains the following specific fields:

- question: A user query entered as a RAG pipe.
- answer: The answer generated from the RAG pipe.
- contexts: Contexts retrieved from external knowledge sources to answer this question.
- ground_truths: Basic factual answers to questions, the only manual annotated information.

B. Experimental setup

To comprehensively evaluate our optimized RAG system, we conducted a comparison experiment to compare the performance of the original LangChain RAG with our proposed RAG optimized by retrieval and reordering.

- Baseline model: LangChain original RAG.
- Comparative experimental design: the output results of the two are systematically evaluated to ensure the consistency of experimental Settings.

C. Evaluation Metric

In this experiment, the following four key evaluation indicators were used to quantitatively score the effect of RAG system:

1) Faithfulness

The faithfulness metric measures the factual consistency of the generated answer against the given context. It is calculated from the answer and retrieved context. The answer is scaled to the (0,1) range. Higher the better. The calculation formula is shown in formula (14):

$$\text{Faithfulness score} = \frac{|\text{Number of claims that can be inferred from given context}|}{|\text{Total number of claims in the generated answer}|} \quad (14)$$

2) Answer Relevancy

Answer Relevancy metric focuses on assessing how pertinent the generated answer is to the given prompt. A lower score is assigned to answers that are incomplete or contain redundant information and higher scores indicate better relevancy. The calculation formula is shown in formula (15):

$$\text{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N} cos(E_{g_i}, E_o) \quad (15)$$

Where:

E_{g_i} is the embedding of the generated question.

E_o is the embedding of the original question.

N is the number of generated questions, which is 3 default.

3) Context Precision

Context Precision is a metric that measures the proportion of relevant chunks in the retrieved contexts. It is calculated as the mean of the precision@k for each chunk in the context. Precision@k is the ratio of the number of relevant chunks at rank

k to the total number of chunks at rank k. The calculation formula is shown in formula (16):

$$\text{Context Precision@k} = \frac{\sum_{k=1}^{K} (\text{Precision@k} \times v_k)}{\text{Total number of relevant items in the top } K \text{ results}} \quad (16)$$

Where K is the total number of chunks in retrieved_contexts and $v_k \in \{0, 1\}$ is the relevance indicator at rank.

4) Context Recall

LLM Context Recall is computed using user_input, reference, and the retrieved_contexts, and the values range between 0 and 1, with higher values indicating better performance. The calculation formula is shown in formula (17):

$$\text{context recall} = \frac{|\text{GT claims that can be attributed to context}|}{|\text{Number of claims in GT}|} \quad (17)$$

D. Experimental Result

1) Precision and recall

Table A shows the comparison of precision and recall before and after optimization. Precision rate focuses on the quality of the search results and measures the proportion of the retrieved documents that are relevant to the user's query, and recall as the proportion of correctly retrieved relevant text to all potentially relevant text. Precision@k is a variant of the precision rate that considers only the top k documents in the search results.

TABLE I. RETRIEVAL ACCURACY AND RECALL RATE

Models	Precision @5 (%)	Recall Rate (%)
LangChain original RAG	75.2	68.5
Optimized RAG	82.7	75.9

It can be seen from Table I that the RAG system optimized by retrieval and reordering has improved the precision by 9.9% and the recall rate by 10.8%. This shows that the optimization strategy can effectively improve the performance of the system in the retrieval stage, and can obtain the relevant context more accurately.

2) Reordering effect evaluation

Table B shows the scores of various evaluation indicators during the reordering stage. We combined model scores, similarity measures, and domain-specific features for a comprehensive score.

TABLE II. RESULTS BASED ON THE RAGAS ASSESSMENT FRAMEWORK

Models	Faithfulness	Answer Relevancy	Context Precision	Context Recall
LangChain original RAG	0.72	0.68	0.70	0.65
Optimized RAG	0.85	0.80	0.78	0.72

According to the results in Table II, the optimized RAG system showed significant improvement in all evaluation indexes, especially in terms of Faithfulness and Answer Relevancy which increased by 18.5% and 17.6%, respectively.

E. Comparative analysis of experimental results

To comprehensively evaluate the effectiveness of the optimization, we compared the overall performance difference between the original LangChain RAG and the optimized RAG system. Table C summarizes the performance of the two models on key indicators.

TABLE III. COMPARATIVE ANALYSIS OF EXPERIMENTAL RESULTS

Models	Overall score (%)
LangChain original RAG	70.3
Optimized RAG	79.0

The data in Table III shows that the optimized RAG system improves the overall score by 12.4%, which verifies the effectiveness of our proposed retrieval and reordering strategy.

V. CONCLUSIONS

This paper introduces a method to improve the accuracy of medical information retrieval with hot keyword vectors in vector databases. Experimental results show that RAG system has significantly improved in all aspects of retrieval and reordering through keyword-enhanced semantic retrieval method and multi-dimensional similarity metric reordering algorithm. It provides strong support for the accuracy and reliability of large model-generating hot topics.

REFERENCES

[1] He J H, et al. "Cutting-edge research and innovative application of large language model in the medical field." Journal of Medical Informatics, vol. 45, no. 9, pp. 10-18, 2024.

[2] Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." arXiv preprint arXiv:2302.13971, 2023.

[3] Yang, An, et al. "Qwen2 technical report." arXiv preprint arXiv:2407.10671, 2024.

[4] Team, Gemini, et al. "Gemini: a family of highly capable multimodal models." *arXiv preprint arXiv:2312.11805*, 2023.

[5] Team, Gemma, et al. "Gemma: Open models based on Gemini research and technology." *arXiv preprint arXiv:2403.08295*, 2024.

[6] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive NLP tasks." Advances in Neural Information Processing Systems, vol. 33, pp. 9459-9474, 2020.

[7] Zhao, Penghao, et al. "Retrieval-augmented generation for ai-generated content: A survey." *arXiv preprint arXiv:2402.19473*, 2024.

[8] Douze, Matthijs, et al. "The Faiss library." arXiv preprint arXiv:2401.08281, 2024.

[9] Zhou Yang, CAI Peihan,and Dong Zhenjiang." Large model Knowledge management System." ZTE Communications Technology, vol. 30, no. 2, pp. 63-71, 2024.

[10] Douze, Matthijs, et al. "The Faiss library." arXiv preprint arXiv:2401.08281, 2024.

[11] Sun, Weiwei, et al. "Is ChatGPT good at search? Investigating large language models as re-ranking agents." *arXiv preprint arXiv:2304.09542*, 2023.

[12] Li, Chaofan, et al. "Making large language models a better foundation for dense retrieval." *arXiv preprint arXiv:2312.15503*, 2023.

[13] Zhang, Peitian, et al. "Retrieve anything to augment large language models." *arXiv preprint arXiv:2310.07554*, 2023.

[14] Qin, Zhen, et al. "Large language models are effective text rankers with pairwise ranking prompting." *arXiv preprint arXiv:2306.17563*, 2023.