

BHT-RAG: Birch-based Hierarchical Tree RAG in Mass Spectrometry

Feilong Wang^{1,2}, Donghui Shi^{1,2*}, Xinyi Cui¹

¹ School of Electronics and Information Engineering, Anhui Jianzhu University, Hefei 230601, China

²Anhui Provincial Joint Key Laboratory for Mass Spectrometry Key Technology R&D and Clinical Application, Hefei 230601, China

Keywords: Mass spectrometry; Large language model; Retrieval-augmented generation; Birch; Chunk

Citation: Wang F.L., Shi D.H., Cui X.Y.: BHT-RAG: Birch-based Hierarchical Tree RAG in Mass Spectrometry. *Data Intelligence*, 2026, Vol. 8, Art. No.: 20250047. DOI: <https://doi.org/10.3724/2096-7004.di.2025.0047>

ABSTRACT

This study presents a Birch-based hierarchical tree retrieval-augmented generation method (BHT-RAG) to mitigate the limitations of information retrieval insufficiency and retrieval quality in the retrieval-augmented generation (RAG) approach within mass spectrometry. Specifically, the documents were initially divided into chunks, and each chunk was then vectorized, clustered, and summarized utilizing Birch and a language generation model. This hierarchical clustering culminated in a recursive hierarchical search tree. To evaluate RAG in the mass spectrometry domain, MSeval, a publicly available RAG evaluation dataset specific to medical mass spectrometry, was further constructed and released. MSeval involves documents for retrieval, question-answer pairs, multiple-choice questions, and fill-in-the-blank questions. In the experimental section, various combinations of clustering methods and generators were initially assessed using the mass spectrometry RAG dataset to gauge their generalization across different tasks, and the optimal generator and clustering method were selected. Subsequently, under the same generator conditions, four other RAG retrieval methods were compared. The experimental results indicate that in the mass spectrometry dataset, BHT-RAG achieves BLEU-4, ROUGE-L, and METEOR scores of 16.7366, 31.1675, and 40.8344 on question-answer tasks; precision, recall, and F1 scores of 90.19%, 89.67%, and 89.93% on multiple-choice tasks; and a maximum accuracy of 80.20% on fill-in-the-blank tasks. BHT-RAG demonstrates superior model generalization across all three tasks compared to other models, showcasing certain effectiveness. Additionally, supplementary ablation experiments, both with and without the clustering module, were conducted to validate its impact. Finally, to maintain

*Corresponding author: Donghui Shi (E-mail: shidonghui@ahjzu.edu.cn; ORCID: 0000-0001-5301-368X).

experimental integrity, tests were conducted on two other publicly available RAG datasets, Wikieval and L-eval, yielding results that corroborate the effectiveness of the proposed approach.

1. INTRODUCTION

In modern medical research, mass spectrometry has been extensively applied across various fields due to its high sensitivity and accuracy, particularly in biomarker discovery, disease diagnosis, and personalized treatment [1-2]. Mass spectrometry determines the composition and quantity of compounds in a sample by measuring the mass-to-charge ratio of ions [3]. Mass spectrometry data, characterized by high dimensionality, complexity, and noise, present difficulties for efficient information retrieval and storage. This highlights the requirement for sophisticated computational tools to process, interpret, and organize such data, facilitating the extraction of valuable insights and bolstering decision-making in medical and biological research.

With the continuous advancement of technology, particularly in recent decades, artificial intelligence has experienced unprecedented development, expanding its research scope to multiple fields. Against this backdrop, Retrieval-Augmented Generation (RAG) technology has gradually emerged as an innovative approach in large-scale text retrieval due to its outstanding performance in natural language processing and information extraction [4]. RAG is a retrieval framework that extracts relevant information from large databases and generates precise answers based on the retrieved results. This technology enhances the accuracy of question-answering systems and facilitates the generation of targeted contents, making the interpretation of complex data and information extraction more intuitive [5]. However, Naive RAG techniques typically rely on direct text chunk retrieval, presenting limitations when applied to medical mass spectrometry data, characterized by its extensive, complex, and variable nature. Relying solely on chunk-based retrieval may result in information loss or redundancy, thereby hindering the effective application of the hierarchical structure of mass spectrometry data. Additionally, while the Naive RAG approach is straightforward, it has inherent drawbacks. Specifically, during text chunking, Naive RAG frequently overlooks contextual information between paragraphs, shows poor organization among chunks, and lacks logical coherence among semantically similar paragraphs. These deficiencies are exacerbated with larger document corpora, leading to diminished retrieval quality and efficiency [6].

To address the aforementioned issues of poor organization among chunks, methods utilizing clustering and summarization have been proposed to reorganize and structure chunks. In response to the fact that existing methods can only retrieve a limited number of contiguous chunks in the corpus, Sarthi et al. [7] proposed using GMM clustering for the embedded chunks, followed by summarization of the clusters. Simon et al. [8] proposed using K-Means clustering on the vector database and employing the Mistral 7B model for summarization to address the constraint of limited context windows supported by large language models. For global retrieval in graph structures, Edge et al. [9] developed a method involving entity relationship extraction with LLMs to construct knowledge graphs (KG) and utilized the Leiden algorithm for community detection. Despite the significant improvements of these approaches in retrieval efficiency and text organization, the mass spectrometry domain still presents unique challenges. Mass

spectrometry data typically contain highly complex, high-dimensional features, such as drug molecular structures, pharmacokinetic information, and drug-related targets. These features impose higher demands on precision and processing capabilities during simple integrated clustering.

Existing RAG approaches often excel in organizing structured data or generating precise localized outputs, yet tend to underperform in maintaining balanced performance across diverse data tasks. This imbalance limits their effectiveness in scenarios demanding both precise contextual understanding and the ability to generalize across complex datasets. To address this gap, the Birch-based hierarchical tree retrieval-augmented generation (BHT-RAG) method has been designed to enhance retrieval quality in a balanced manner, aiming to improve performance across multiple dimensions of information processing.

Herein, a BHT-RAG was proposed to optimize the clustering and retrieval-generation process for mass spectrometry data, which enhanced the efficiency and quality of text information retrieval. Birch was employed for recursive clustering and summarization, long texts were segmented and a tree structure was constructed. Parent nodes were utilized to represent cluster information, facilitating efficient and accurate text organization and retrieval. This approach offers an effective solution for high-quality information retrieval.

The main contributions of this paper are as follows:

1. A Birch-based hierarchical tree retrieval-augmented generation method (BHT-RAG) was proposed as a novel approach for text retrieval and generation. This method employs recursive clustering and language generation models to summarize information and efficiently organize large-scale text data into a tree structure, thereby enhancing retrieval quality in mass spectrometry.

2. A RAG evaluation dataset (MSeval) for the medical mass spectrometry domain was constructed and made publicly available, providing reliable benchmarking data for subsequent medical data retrieval and generation tasks (<https://github.com/wfl0325/Medical-Mass-Spectrometry-Dataset>). The dataset includes multiple-choice questions, fill-in-the-blank questions, Question-Answering pairs and documents required for retrieval, providing standardized data support for model evaluation and performance comparison in subsequent studies.

3. A series of comparative experiments were conducted to evaluate the performance of diverse clustering algorithms and generator models within the BHT-RAG framework. The optimal clustering method and generator model were thereby identified. Furthermore, ablation experiments and tests on additional public datasets were conducted to provide additional evidence for the necessity and effectiveness of the proposed method.

2. RELATED WORK

Mass spectrometry technology has become a vital tool in biomedical research, efficiently analyzing and detecting the types and quantities of biomolecules. It has found significant applications in fields such

as disease biomarker identification and drug metabolism studies [10]. However, mass spectrometry data typically exhibit high dimensionality, nonlinearity, and heterogeneity, posing challenges for traditional data analysis and information extraction methods in accurately retrieving information from large-scale datasets. Specifically, mass spectrometry data frequently contain substantial noise and complex feature distributions, which further complicate the retrieval and analysis processes.

The deployment of large pre-trained language models (LLMs) has markedly enhanced the system's capacity for natural language processing [11-13]. These models demonstrate robust In-context learning [14] and emergent [15] abilities. Conventional large models, despite their generative and reasoning prowess, exhibit significant limitations in practical applications, particularly in ensuring the timeliness and completeness of knowledge. The training data for LLMs are primarily derived from publicly available online information, which hinders their ability to access real-time, highly confidential, or specialized data [16-18]. This has led to the generation of inaccurate or incomplete responses when dealing with questions that required domain-specific or real-time information. Furthermore, these models are susceptible to "hallucination" [19], a phenomenon where generated responses seem plausible but are factually incorrect. The fundamental issue could be attributed to the model's dependency on statistical and probabilistic techniques for text generation, as opposed to genuine knowledge reasoning. Consequently, when confronted with unfamiliar domains, these models are likely to generate misleading outputs. In practical applications, the dissemination of misleading information could lead to significant adverse consequences, particularly in domains such as healthcare and law, where erroneous data could result in irreparable damage.

To address these challenges, RAG has been developed to enhance system performance in question-answering and task-processing by integrating retrieval techniques with generative models, which facilitates the extraction of relevant information from private databases or other external data sources while leveraging the generative capabilities of language models. Contrasting with traditional generative models, RAG identifies the documents most relevant to a given query through a retrieval module and subsequently inputs these retrieved results into the generative model alongside the query to produce the final answer [20]. By injecting external retrieval results into the prompt, RAG can dynamically expand the knowledge range of the language model [21]. The core concept of RAG involves retrieving relevant knowledge from external data sources via a retrieval module in response to a user query. This information is subsequently integrated into the prompt template of the language model, enabling the generative model to produce the final answer.

Lewis et al. [4] proposed a framework combining information retrieval with the generative capabilities of large language models. The concept of RAG has been shown to outperform other methods in a number of tasks, particularly those that require the integration of external knowledge, such as open-domain question-answering (Open-Domain QA) [22]. Specifically, Kreimeyer et al. [23] addressed the inefficiency of expert literature review in precise oncology treatment discussions by proposing a method that integrated an RAG pipeline with large language models. This method retrieves pertinent textual data from reliable data sources and generates responses, thereby reducing the burden of manual literature review. By

employing the non-fine-tuned Llama 2 model for the retrieval of pertinent summaries and responses to queries, this methodology successfully reproduces over 80% of treatment relationships, addressing the time-consuming issue of manual literature review. However, this method relies heavily on the quality of the retrieved summaries and lacks mechanisms to ensure coverage of less-studied treatment relationships, potentially resulting in critical information gaps.

Jiangtong et al. [24] put forth the RA-CFGPT system as a means of addressing the demands of question-answering, document analysis, and other tasks in Chinese financial contexts. The system incorporated a hybrid knowledge base, a fine-tuned Chinese financial large language model (FinLLM), an information organizer, and a response checker, thereby ensuring the accuracy, compliance, and risk-awareness of outputs. The experimental results demonstrated the effectiveness of retrieval-augmented training and the RA-CFGPT system in addressing the issues of accuracy and compliance in financial tasks. Nonetheless, the model's performance is domain-specific, which may impede its ability to generalize to other financial or non-financial domains, thus limiting its broader applicability. To address the challenges of noisy retrieved nodes and the absence of interactive representations between questions and graph entities in commonsense question-answering, Sha et al. [25] proposed the Retrieval-Augmented Knowledge Graph (RAKG) model, which corrected erroneous knowledge paths through a density matrix, extracted augmented subgraphs, and integrated question and entity representations using a bidirectional attention strategy. Experimental results demonstrated that RAKG markedly enhanced the accuracy of responses on the CommonsenseQA and OpenBookQA datasets. However, the reliance on pre-constructed knowledge graphs and complex entity extraction pipelines could still introduce scalability challenges when applied to domains with sparse or incomplete knowledge graphs.

The Naive RAG system follows a traditional three-step process of indexing, retrieval, and generation [26]. During the indexing stage, the system transforms raw data from various formats into plain text, segments it into manageable pieces, and encodes these pieces into vectors using an embedding model. These vectors are stored in a vector database to enable subsequent similarity searches. In the retrieval stage, user queries are also vectorized, and the system calculates similarity scores to identify the most relevant text chunks from the database. In the generation stage, the system combines the user query with the retrieved text chunks to form a prompt, which is then utilized by a large language model to generate a response.

To overcome the limitations of Naive RAG methods, advanced RAG approaches have introduced pre-retrieval and post-retrieval strategies aimed at improving the quality and relevance of retrieved information. In the pre-retrieval stage, techniques such as fine-grained segmentation, metadata incorporation, query rewriting, and mixed retrieval methods are employed to optimize the indexing structure and enhance the clarity of user queries [26-29]. Post-retrieval strategies, including reranking retrieved chunks and compressing context, focus on mitigating information overload by emphasizing critical sections and ensuring that large language models process merely the most relevant information. However, these approaches often involve significant computational overhead, especially for large datasets, and may struggle to maintain high retrieval efficiency in real-time applications.

Furthermore, modular RAG architectures have been developed to facilitate the adaptability and flexibility [30]. These frameworks introduce new modules, such as search modules for similarity-based searches [31], memory modules for iterative self-enhancement [32-33], and task adapters tailored for specific downstream tasks [34-35]. Modular RAG systems also allow for dynamic adjustments in module arrangement and interaction flow, such as the iterative Retrieve-Read-Retrieve-Read framework and the Demonstrate-Search-Predict (DSP) framework [36]. These innovations address specific challenges in retrieval tasks and allow RAG systems to integrate seamlessly with other technologies, such as reinforcement learning and fine-tuning [37-38], enhancing both retrieval accuracy and the ability to generalize across diverse datasets. Nevertheless, the modular design introduces complexities in system integration and optimization, potentially increasing implementation costs and necessitating expertise for effective deployment.

The development of RAG has prompted a shift in retrieval methods, moving away from term frequency-based approaches such as BM25 [39] and TF-IDF [40] towards deep learning-based embedding representations. This shift aims to enhance retrieval accuracy [41-42]. Specifically, Lan et al. [43] proposed a method termed CoG based on gradually copying text segments and addressed the issue of existing text generation models relying on a fixed vocabulary for word-by-word generation. This approach addressed issues pertaining to generation efficiency and domain adaptability. By calculating the contextual representations of text segments and employing efficient vector search tools for indexing, the text generation task was simplified into a series of copy-and-paste operations. The experimental results demonstrated that this method outperformed traditional models in terms of generation quality on the WikiText-103 benchmark, while achieving inference efficiency comparable to autoregressive models. Du et al. [44] proposed the concept of R-GQA (Retrieval-Augmented Generative Question-answering) as a means of addressing event parameter extraction. R-GQA utilized a retrieval process to identify the most similar question-answer pairs, which were then used as prompts alongside the current example context. The parameters were subsequently decoded into answers. This method demonstrated superior performance compared to previous approaches across diverse settings, including fully supervised learning, domain transfer, and few-shot learning.

Research has concurrently explored methods to boost RAG performance in complex dialogues, including optimizing prompt engineering in the generation module and integrating multi-turn dialogue history. Specifically, Guo et al. [45] highlighted the inadequate deployment of retrieval-augmented techniques in non-knowledge-intensive (NKI) tasks and proposed a dual-stage framework known as PGRA. This framework addressed the necessity for diverse relevance scoring functions while maintaining a balance between training costs and task performance. The initial stage involved constructing a shared static index with a task-agnostic retriever to efficiently select candidate evidence. The subsequent stage involved designing a prompt-guided re-ranker for re-sorting the evidence in accordance with task-specific relevance. The experimental results demonstrated that PGRA outperformed other state-of-the-art retrieval-augmented methods in NKI tasks and explored the factors influencing model performance and its generalizability. To tackle the prevalent ambiguity in open-domain question-answering, Kim et al. [46] proposed the ToC framework, which accommodated questions with multiple potential interpretations.

ToC employs a recursive approach to construct a disambiguation tree for ambiguous questions, utilizing few-shot prompts and external knowledge to generate comprehensive answers. The experimental results demonstrated that ToC exhibited superior performance compared to existing baseline models in the few-shot setting of ASQA. It surpassed full supervised baselines utilizing the complete training set on both Disambig-F1 and Disambig-ROUGE metrics. Furthermore, the optimization of RAG encompassed retrieval unit optimization [47-49], index optimization [50], and query rewriting [51-52].

Despite the rapid development of RAG technology, further exploration is still needed to determine the most effective processing and utilization of text chunks within RAG systems.

3. BHT-RAG METHOD

Figure 1 illustrates the BHT-RAG method. This method was divided into two main parts: constructing the tree in a bottom-up manner and retrieving from the tree in a top-down manner. In the tree construction phase, the method initially segmented the text dataset, dividing long texts into text chunks that preserved complete semantic information. Following that, these chunks were grouped into multiple semantically similar clusters utilizing the Birch clustering algorithm, and a large language model summarizing each cluster was employed to create a parent node representing the cluster information. The process involved recursively clustering and summarizing until a root node set was constructed for the tree structure, representing the core information of the entire text dataset. In the retrieval phase, the method calculated

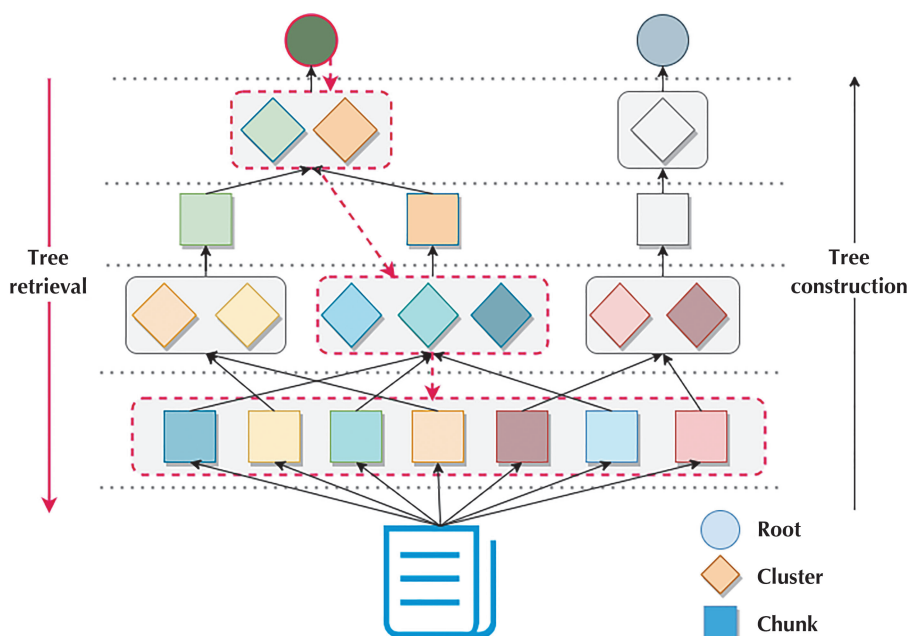


Figure 1. Structure Diagram of the BHT-RAG Method.

the similarity between the query and the root nodes of the tree. A recursive search was conducted to identify the most relevant information path and ultimately generated the final response by combining the parent node information from the path with other block information.

3.1 Tree Construction

The text dataset $\mathcal{D}$ was initially processed by dividing it into chunks (chunking). Specifically, given a dataset $\mathcal{D} = \{T_1, T_2, \dots, T_m\}$, where each segment of text T_i could be split into multiple text chunks $\{c_{i1}, c_{i2}, \dots, c_{in_i}\}$. This process could be described by Equation (1):

$$\mathcal{C} = \bigcup_{i=1}^m \text{Chunking}(T_i) \quad (1)$$

where T_i represents the i -th segment of text, $\mathcal{C}$ represents the set of all text chunks, and $\{c_{i1}, c_{i2}, \dots, c_{in_i}\}$ are the corresponding chunks following the chunking process.

In this study, the tiktoken toolkit, developed by OpenAI, was employed for tokenization during the text processing stage. Tiktoken was designed for efficient token splitting, particularly suitable for handling inputs for large models such as GPT.

Subsequently, each sentence within the text chunks was vectorized to meet the clustering requirements. Let $\mathcal{A}$ denote the clustering algorithm, which divides the set of text chunks $\mathcal{C}$ into k clusters. Each cluster $\mathcal{C}_j$ consists of a group of semantically similar text chunks. The clustering process is shown in Equation (2), with $\mathcal{C}_j$ representing an individual cluster:

$$\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\} = \mathcal{A}(\mathcal{C}) \quad (2)$$

The Birch (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm stands out as an efficient hierarchical clustering method particularly suitable for large-scale datasets. The core concept of Birch lies in incrementally compressing data by constructing a Clustering Feature Tree (CF Tree), allowing the algorithm to perform clustering in linear time while maintaining high accuracy. The Birch algorithm consists of four main phases, namely CF Tree construction, centroid selection, reclustering, and merging.

The CF Tree was built utilizing the Birch algorithm by incrementally scanning the data. During this process, data points were progressively added to the leaf nodes of the tree. The CF Tree serves as a balanced tree structure where each node stores multiple Clustering Features (CFs) that describe local data characteristics. Each CF contains three elements: the number of data points N , the linear sum of the data points $LS = \sum_{i=1}^N x_i$, and the squared sum of the data points $SS = \sum_{i=1}^N x_i^2$. These values were hereby used to calculate the local cluster's centroid μ and radius R , as shown in Equations (3) and (4):

$$\mu = \frac{LS}{N} \quad (3)$$

$$R = \sqrt{\frac{SS}{N} - \left(\frac{LS}{N}\right)^2} \quad (4)$$

Each non-leaf node of the CF Tree stored the clustering features of its subtree, while the leaf nodes contained data points for local clusters. As data were scanned, points were dynamically inserted into leaf nodes, with the CF Tree maintaining balance and controlling node radii to prevent excessive depth or width. After constructing the tree, the Birch algorithm performed an initial analysis of these local clusters based on their centroids and further refined the clustering. Finally, Birch reduced the total number of clusters by merging those within a predefined distance threshold.

Following the clustering of text chunks, a large language model was employed to further summarize the information within each cluster $\mathcal{C}_j$, generating the corresponding parent node P_j . This parent node could be viewed as the semantic center of all the text chunks within the cluster, representing the core content of the cluster. The process of generating the parent node could be described by Equation (5):

$$P_j = \text{Summary}(\mathcal{C}_j) \quad (5)$$

At this point, the set of parent nodes at the current level $P = (P_1, P_2, \dots, P_k)$ had been obtained. To ensure the hierarchical structure of the tree and maintain its simplicity, the next step involved determining the size of this parent node set, i.e., to calculate the number of parent nodes at the current level, denoted as $|P|$, as shown in Equation (6). When the size exceeded the predefined threshold n , the current set of parent nodes could be treated as a new set of text chunks. The clustering and summarization processes were then repeated to generate a new set of parent nodes, continuing recursively until the number of parent nodes at a given level was less than or equal to the threshold n . This process could be viewed as recursively compressing and refining the information until a parent node set suitable for constructing the root node of the tree was obtained.

$$|P| = \mathcal{A}(P) \quad (6)$$

Upon the completion of the recursive process, the final set of parent nodes became the root node set of the tree structure, representing the core information of the entire text dataset. The algorithm for constructing the hierarchical tree in BHT-RAG is shown in Algorithm 1.

Algorithm 1. The algorithm for constructing the hierarchical tree in BHT-RAG.

Input: Document set $\mathcal{D} = \{T_1, T_2, \dots, T_m\}$

Output: The tree structure $\mathcal{T}$ consists of both leaf nodes and root nodes.

1. $c_{in_i} \leftarrow \text{Chunking}(T_i)$
 2. **For** each c_{in_i} **do**
 3. $\text{Embedding}(c_{in_i}) \rightarrow e_{c_{in_i}}$
 4. $Cl_j \leftarrow \mathcal{A}(e_{c_{in_i}})$
 5. **While** $|Cl_j| > n$
 6. **For** each Cl_j **do**
 7. $P_j \leftarrow \text{Summary}(Cl_j)$
 8. $\text{Embedding}(P_j) \rightarrow e_{P_j}$
 9. $Cl'_j \leftarrow \mathcal{A}(e_{P_j})$
 10. **If** $|Cl'_j| < n$ **do**
 11. Break
 12. Return $\mathcal{T}$
-

3.2 Tree Retrieval

Herein, during the retrieval process of the hierarchical tree, BHT-RAG first calculated the similarity between the query and the root nodes of the tree. For each root node P_j , the cosine similarity between the query Q and the node was computed, and the parent node P_{max} with the highest similarity was selected as the starting point for the next level of search. The similarity calculation is shown in Equation (7).

$$\text{Sim}(P_j, Q) = \frac{P_j \cdot Q}{\|P_j\| \|Q\|} \quad (7)$$

Subsequently, this search process was conducted recursively from the root node to the leaf nodes, the similarity between the user query and parent nodes at each level was calculated, and the most relevant path was selected. Ultimately, by combining the information from the highest-similarity parent nodes at each layer, all related summary text blocks and initial text blocks were extracted to generate the final response. This integration of refined text blocks enhanced the comprehensiveness of the retrieved information and rendered the semantic representation more precise and nuanced.

4. EXPERIMENT

4.1 Model Selection and Dataset Construction

In the experiments, four distinct clustering algorithms, namely hierarchical clustering, GMM clustering, Birch clustering, and spectral clustering, were selected for evaluation. Furthermore, three representative models, namely GPT-3.5-turbo, GLM-4, and Qwen2-7B-chat, were selected for comparative analysis to evaluate the performance of large language models across a range of tasks.

In order to address the lack of medical mass spectrometry-related datasets in the field of RAG, MSeval, a specialized dataset for RAG retrieval and evaluation, was hereby constructed. This dataset encompasses three task dimensions, including multiple-choice questions, fill-in-the-blank questions, and question-answer pairs. This comprehensive assessment was conducted to evaluate the generalization capabilities of the model. A dataset in the JSON format, comprising 7,650 drugs, was hereby collated from online sources. Each drug was represented by at least 80 attributes, including an overview, molecular formula, metabolic pathway, and half-life. In order to satisfy the requirements of RAG methods, the attribute information was aggregated and integrated into coherent sentences. Specifically, a sentence was constructed for the drug Dactinomycin based on its “overview” and “metabolic pathway” attributes. Dactinomycin is an actinomycin employed for the treatment of a range of cancers, with the liver being its primary metabolic pathway. Similarly, text documents were generated in a readily readable format for different drugs and their attributes, thus providing a data source for retrieval. The generation of the question-answer pair dataset was conducted in accordance with the aforementioned method, whereby all drugs and their attributes were iterated through to create corresponding question-answer pairs for the readable sentences. The multiple-choice question dataset was constructed by designating the true attribute value of a drug as the correct answer and randomly selecting three incorrect attribute values as distractors. Similarly, the fill-in-the-blank questions were generated by traversing and combining drug attributes to generate completion prompts.

The dataset has been made publicly available on GitHub for use in future research. Furthermore, to gain insight into the influence of token count on retrieval performance, the documents were hereby classified into three categories based on the number of tokens for experimental purposes. The initial document contained 39,049 tokens, which were subsequently reduced to 23,975 and 14,418 tokens, respectively, to assess the model’s performance at varying granularities. The documentation groups are shown in Table 1.

Table 1. Text Divided into Different Groups.

Group	Token Count
Group 1	39,049
Group 2	23,975
Group 3	14,418

4.2 Experimental Metrics

In order to evaluate the model's capacity for generalization, various metrics were employed for each of the three tasks.

In this study, for the question-answering task, the evaluation metrics employed were BLEU-4, ROUGE-L, and METEOR. BLEU primarily measured the degree of n-gram overlap between the generated text and the reference text. BLEU-4, on the other hand, was specifically designed for the evaluation of sentences utilizing four-gram (4-gram) sequences. The calculation formula for BLEU-4 is presented in Equation (8):

$$BLEU = BP \times \left(\sum_{n=1}^N w_n \log p_n \right) \quad (8)$$

The BP penalty is generally applied in cases involving too brief generated sentence. The p_n represents the n-gram precision between the generated text and the reference text, while the w_n denotes the weight assigned to each n-gram.

The BLEU-4 metric is primarily employed for the assessment of the accuracy of short texts. However, it is not an effective means of capturing the semantics and structure of sentences. In contrast, ROUGE-L is based on the longest common subsequence (LCS) and measures the similarity between the generated and reference texts in terms of sequence alignment, focusing on the structural alignment of generated sentences. The calculation process is illustrated in Equation (9):

$$ROUGE - L = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \quad (9)$$

In Equation (9), *Precision* can be defined as the proportion of the LCS length in the generated text that matches the reference text, expressed as a percentage of the total length of the generated text. The *Recall* indicator represents the proportion of the LCS length in the generated text that corresponds to the reference text, expressed as a percentage of the total length of the reference text. The β parameter is employed to adjust the relative importance of *Precision* and *Recall*, with a typical value of $\beta=1$. ROUGE-L places emphasis on the matching of word order and text structure, rendering it an appropriate metric for evaluating generation tasks.

METEOR is an evaluation metric specifically designed for the assessment of translation and generation tasks. It is based on lexical matching, taking into account a number of factors, including word inflections, synonym matching, and the relative importance of word order. The calculation of METEOR is shown in Equation (10).

$$METEOR = F_{mean} \times (1 - Penalty) \quad (10)$$

where F_{mean} is the weighted harmonic mean of *Precision* and *Recall*, as shown in Equation (11):

$$F_{mean} = \frac{10 \cdot Precision \cdot Recall}{Recall + 9 \cdot Precision} \quad (11)$$

Penalty represents a penalization factor reflecting the difference in word order between the generated and reference text, calculated as in Equation (12):

$$Penalty = 0.5 \times \left(\frac{chunks}{matches} \right)^3 \quad (12)$$

where *chunks* represents the number of contiguous matched segments between the generated and reference text, and *matches* refers to the number of matching words.

METEOR outperforms BLEU and ROUGE in handling inflectional variations and semantic similarity, rendering it a more adaptable choice for natural language generation tasks.

In the evaluation of fill-in-the-blank tasks, the primary focus is on the model's accuracy in predicting the correct answers given the provided context. Accuracy serves as the core evaluation metric for fill-in-the-blank tasks, measuring the degree of alignment between the model's output and the reference answers. The specific definition is provided in Equation (13).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (13)$$

where *TP* represents true positives, *TN* represents true negatives, *FP* represents false positives, and *FN* represents false negatives.

In order to evaluate the performance of a model in a multiple-choice task, three metrics, namely *Precision*, *Recall*, and *F1-score*, could be employed to comprehensively evaluate the model performance. The formulas for these three metrics are presented in Equations (14), (15), and (16), respectively:

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (16)$$

4.3 Experimental Results and Analysis

In order to ensure the randomness and comprehensiveness of the experimental results, 100 multiple-choice questions, 100 fill-in-the-blank questions, and 100 question-answer pairs were hereby randomly selected from the MSeval for testing. The series of experiments allowed for a more accurate evaluation of the performance of diverse generator models and clustering techniques, thereby forging a foundation for

further enhancement of the RAG system. Furthermore, in the vectorization process, each model employed its respective vectorization interface for word vectorization.

Table 2 presents the scores of Group 1, Group 2, and Group 3 in both the question-answering and multiple-choice tasks. In each group, bolded values indicate the highest score for the respective metric in that row, while underlined values represent the second-highest score, and a “*” symbol highlights the highest metric across the entire table.

For the question-answering task, the maximum BLEU-4, ROUGE-L, and METEOR scores in Group 1 were 16.7366, 31.1675, and 40.8344, respectively, achieved by the combination of GPT-3.5 and Birch clustering. Additionally, most of the highest metrics across models were concentrated when using the Birch clustering method, with only the METEOR score in the GLM4 model and the BLEU-4 score in Qwen2-7B-chat model reaching their peak values using spectral clustering. In Group 2, the highest BLEU-4, ROUGE-L, and METEOR scores were 12.2364, 25.6392, and 28.1171, respectively, generated by GPT-3.5-turbo paired with GMM. Qwen2-7B-chat performed the weakest across all clustering methods, with peak scores of BLEU-4, ROUGE-L, and METEOR reaching only 8.1948, 20.3087, and 24.8801, which were 4.0416, 5.3305, and 3.2370 below the top scores, respectively. The GLM4 model’s performance fell between the other two models, second only to GPT-3.5-turbo. In Group 3, a significant drop in all evaluation metrics was observed, with the maximum values being 8.7186, 21.6494, and 23.5965, respectively. The analysis revealed that reducing document length concurrently led to a decrease in question-relevant content, causing a scarcity of retrievable information. Consequently, the model had to heavily depend on its internal knowledge base, which in turn caused a significant divergence between the model’s responses and the reference answers, thereby contributing to the overall reduction in metric values.

For the multiple-choice task, precision, recall, and F1 score metrics were employed to evaluate model generalization, with macro-averaging applied across all the three metrics. GLM4 demonstrated superior performance among the models, achieving 95.80% for precision, 95.78% for recall, and 95.78% for F1 score. For precision, Qwen2-7B-chat ranked second, reaching 95.26%, despite a notable decrease in its recall value. The GPT-3.5-turbo model exhibited slightly lower performance on specific metrics compared to other models, yet its overall stability was maintained. In Group 2, the GLM4 model maintained the highest scores across metrics, with most maximum values exceeding 94.00%. Meanwhile, the Birch clustering method demonstrated particularly strong results. For precision, the Qwen2-7B-chat model closely followed GLM4, achieving 92.94%.

Additionally, spectral clustering in Group 2 also exhibited impressive performance, often achieving the highest or second-highest values across different models. The GPT-3.5-turbo model remained stable, with a precision score slightly lower compared to Qwen2-7B-chat. However, its recall and F1 scores either matched or exceeded those of Qwen2-7B-chat. In Group 3, contrasting with the situation in previous groups, the hierarchical clustering method, previously less effective, showed remarkable performance, achieving precision, recall, and F1 scores of 90.76%, 89.77%, and 90.02% when paired

Table 2. Performance of Different Models and Clustering Methods in Q&A Tasks and Multiple-Choice Questions.

Task	Model	Metric	Group 1				Group 2				Group 3			
			Birch	Hierar-chical	GMM	Spectral	Birch	Hierar-chical	GMM	Spectral	Birch	Hierar-chical	GMM	Spectral
QA	GLM4	BLEU-4	13.4964	12.3559	<u>12.6127</u>	12.5810	<u>10.3523</u>	10.3593	9.5238	9.1703	7.5888	6.2467	<u>7.4467</u>	6.7925
		ROUGE-L	27.1180	26.0783	<u>26.5158</u>	25.7247	21.9027	<u>21.7145</u>	21.3746	20.2498	<u>18.2425</u>	17.4399	18.6985	16.9135
		METEOR	35.8375	34.7651	36.6678	<u>35.9380</u>	<u>28.3462</u>	28.4426	28.3313	27.0606	<u>24.3322</u>	23.7309	24.8755	24.1864
		BLEU-4	16.7366*	<u>15.6123</u>	14.8729	15.4743	<u>11.9700</u>	11.8773	12.2364	11.5724	8.7186	7.9559	<u>8.6940</u>	7.6944
		ROUGE-L	31.1675*	30.7498	30.3515	<u>30.9600</u>	24.3400	24.6247	25.6392	<u>25.3468</u>	21.6494	20.1480	20.8506	<u>20.8522</u>
Choice	GPT-3.5-turbo	METEOR	40.8344*	<u>40.3725</u>	38.8034	39.1670	26.7400	<u>27.5277</u>	28.1171	27.4723	23.5965	22.3842	22.6279	<u>23.0091</u>
		BLEU-4	<u>11.2946</u>	9.0567	9.5831	11.4083	8.1948	7.6544	7.7440	<u>7.7820</u>	6.2165	<u>5.7871</u>	5.4957	4.8614
		ROUGE-L	25.7408	24.4195	23.2640	<u>25.7032</u>	20.3087	19.4997	19.0419	<u>20.1531</u>	17.034	<u>16.7655</u>	16.6831	16.2874
		METEOR	34.1957	32.2984	31.0588	<u>32.4781</u>	24.4718	24.4049	24.8801	<u>24.7341</u>	20.3246	21.9119	<u>21.3547</u>	20.3934
		Precision/%	<u>94.89</u>	94.75	95.80*	93.64	94.99	91.48	93.08	<u>94.01</u>	<u>90.59</u>	90.76	<u>90.59</u>	89.81
Choice	GPT-3.5-turbo	Recall/%	93.35	92.84	95.78*	<u>93.99</u>	94.88	91.63	91.94	<u>93.41</u>	<u>89.20</u>	89.77	<u>89.20</u>	88.30
		F1	<u>94.05</u>	93.70	95.78*	93.70	94.88	91.40	92.40	<u>93.65</u>	89.70	90.02	89.70	88.79
		Precision/%	90.19	<u>91.58</u>	93.78	<u>91.58</u>	87.93	90.29	85.51	<u>89.84</u>	85.39	82.86	<u>83.23</u>	81.59
		Recall/%	89.67	<u>90.53</u>	91.14	<u>90.53</u>	86.89	<u>87.86</u>	84.25	88.40	82.71	80.78	<u>81.14</u>	77.97
		F1	89.83	<u>90.91</u>	92.15	<u>90.91</u>	87.32	<u>88.86</u>	84.81	89.08	83.80	81.77	<u>81.99</u>	79.54
Choice	Qwen2-7B-chat	Precision/%	91.44	92.93	<u>93.96</u>	95.26	92.05	91.56	<u>92.91</u>	92.94	<u>85.17</u>	81.20	81.20	90.75
		Recall/%	<u>88.53</u>	86.89	87.46	88.82	<u>87.15</u>	81.08	83.76	87.71	80.67	72.38	72.38	<u>73.01</u>
		F1	89.71	89.72	<u>90.31</u>	91.65	<u>89.44</u>	85.83	87.96	89.73	82.72	76.16	76.16	<u>80.56</u>

with GLM4. Observations indicated significant variability in the Qwen2-7B-chat model's performance; while its highest precision score reached 90.75%, its recall and F1 scores were only 80.67% and 82.72%, respectively. Analysis of experimental records revealed that this model occasionally yielded incorrect answer choices. For instance, when asked "What is the DrugBank Accession Number of Paclitaxel?" with the correct response being "B: DB01229," the model incorrectly identified "A: DB01229," leading to an erroneous answer.

Table 3 presents the accuracy of different models and clustering methods in the fill-in-the-blank task across all groups. In the table, the bold values indicate the maximum accuracy for each model under the four clustering methods within the same group. The underlined values represent the second-highest accuracy. The symbol "*" denotes the overall maximum accuracy for the corresponding values in the entire table. As shown in the table, the GPT-3.5-turbo model achieved the highest accuracy across all four clustering methods, with the spectral clustering method attaining the highest accuracy of 81.19, followed by Birch clustering with a score of 80.20. The Birch clustering method demonstrated consistent performance in both Group 1 and Group 2, frequently attaining the highest or second-highest accuracy. In contrast, the Qwen2-7B-chat model displayed considerable instability in Group 3, suggesting its limited generalization capabilities when the volume of document information was reduced.

Table 3. Accuracy of Fill-in-the-Blank Tasks Across All Groups/%.

Model	Group					
	1		2		3	
GPT-3.5-turbo	Birch	80.20*	Birch	63.37	Birch	50.49
	Hierarchical	69.31*	Hierarchical	58.42	Hierarchical	48.51
	GMM	78.22*	GMM	56.44	GMM	51.49
	Spectral	81.19*	Spectral	57.43	Spectral	48.51
GLM4	Birch	66.34	Birch	59.41	Birch	53.47
	Hierarchical	65.35	Hierarchical	59.41	Hierarchical	44.55
	GMM	70.30	GMM	58.42	GMM	54.46
	Spectral	63.37	Spectral	59.41	Spectral	49.50
Qwen2-7B-chat	Birch	69.31	Birch	60.40	Birch	43.56
	Hierarchical	62.38	Hierarchical	46.53	Hierarchical	44.55
	GMM	62.38	GMM	55.45	GMM	45.54
	Spectral	63.37	Spectral	54.46	Spectral	44.55

This study evaluated the performance and stability of clustering methods and models by averaging their scores across various tasks, thereby informing the selection process.

Table 4 demonstrates the average scores of the different clustering methods for the three test evaluations, where the average score for the multiple-choice questions represents the average of the sum of precision, recall, and F1. The analysis confirmed the best performance of the Birch clustering method in all assessments, particularly in the scoring of multiple-choice and fill-in-the-blank tasks. In the question-answer pair evaluation, the Birch method achieved the highest score of all the methods, with a total of 20.76. Furthermore, the Birch clustering method yielded the highest scores for multiple-choice and fill-in-the-blank tasks, with an average of 89.15 and 60.72%, respectively.

Table 4. Average Scores of Different Clustering Methods Across Three Evaluations.

Method	Avg		
	Question-Answering Pairs	Multiple-Choice Questions	Fill-in-the-Blank Questions/%
Birch	20.76	89.15	60.72
GMM	20.27	88.08	59.18
Spectral	20.14	88.85	57.97
Hierarchical	20.15	87.76	55.44

The GMM clustering method showed a marginal decrease in performance compared to Birch. Despite a lower score of 20.27 in question-answer pairs, it maintained high scores of 88.08 in multiple-choice and 59.18% in fill-in-the-blank questions. The GMM method exhibited stability in question-answer pairs and multiple-choice questions, presenting certain efficacy in tasks with clearly defined answers. However, its performance in fill-in-the-blank questions demonstrated a slight decline. The Spectral clustering method exhibited generally average performance, involving a score of 20.14 in question-answer pairs, which was slightly lower than that of the previous two methods. The score achieved in multiple-choice questions was 88.85, which was comparable to that of GMM. However, its performance in the fill-in-the-blank section was markedly inferior, scoring 57.97%, significantly lower than Birch and GMM. The hierarchical clustering method also performed poorly, achieving a score of 20.15 in question-answer pairs, similar to that of the spectral clustering method. The lowest scores were observed for this method in both the multiple-choice and fill-in-the-blank questions, involving scores of 87.76 and 55.44%, respectively.

In general, the Birch clustering method demonstrated comprehensive superiority across all three evaluation tasks, particularly in multiple-choice and fill-in-the-blank questions, rendering it the optimal choice for clustering methods.

Table 5 demonstrates the performance of various generators across the three evaluation tasks of question-answering, multiple-choice questions, and fill-in-the-blank questions. In the question-answering

task, GPT-3.5-turbo was distinguished by its superior performance, attaining the highest scores on the three generation quality assessment metrics. The BLEU-4, ROUGE-L, and METEOR scores were 25.5566 and 30.0543 for ROUGE-L and METEOR, respectively. GPT-3.5-turbo demonstrated superior capabilities in maintaining semantic relevance and linguistic fluency when generating responses based on retrieved texts. Qwen2-7B-chat's performance was relatively weaker, achieving scores of 7.9232 for BLEU-4, 20.4084 for ROUGE-L, and 26.0422 for METEOR.

Table 5. Average Scores of Different Models Across Three Evaluation Tasks.

Model	Question-Answering			Multiple-Choice Questions			Fill-in-the-Blank Questions
	BLEU-4	ROUGE-L	METEOR	Precision/%	Recall/%	F1/%	Accuracy/%
GLM4	9.8772	21.831	29.3761	92.86	92.02	92.31	58.66
GPT-3.5-turbo	11.9512	25.5566	30.0543	87.81	85.98	86.74	61.96
Qwen2-7B-chat	7.9232	20.4084	26.0422	90.11	82.48	85.82	54.37

In the multiple-choice task, GLM4 exhibited robust performance, attaining the highest scores in the Precision, Recall, and F1 metrics, with an F1 score of 92.31%. In contrast, GPT-3.5-turbo recorded an F1 score of 86.74% in the multiple-choice task, which might be indicative of a potential lag in its knowledge base. While Qwen2-7B-chat demonstrated satisfactory performance in recall, its precision and F1 scores were comparatively lower than those of GLM4 and GPT-3.5-turbo, indicating its relatively weaker overall performance in the multiple-choice task.

In the fill-in-the-blank task, GPT-3.5-turbo demonstrated continued excellence, attaining an accuracy score of 61.96%, the highest among the three models. GLM4 achieved a comparable performance with an accuracy score of 58.66%, slightly lower than GPT-3.5-turbo, yet still indicating satisfactory accuracy. Qwen2-7B-chat demonstrated an accuracy score of 54.37%, suggesting a limited effectiveness in the fill-in-the-blank task and indicating potential difficulties in predicting specific missing information. Figure 2 presents a visual representation of the radar charts for the three models across different tasks.

In light of the preceding experimental outcomes, the GPT-3.5-turbo model generator was hereby integrated with the Birch clustering method for the forthcoming comparative experiments.

4.3.1. Comparative Experiments

In the comparative experiments, BM25, TF-IDF, RAPTOR [7], and the widely used LangChain-Chatchat (<https://github.com/chatchat-space/Langchain-Chatchat>) were selected for testing. For BM25 and TF-IDF, the top 50 retrieval results were adopted, with GPT-3.5-turbo serving as the generation model. In the LangChain-Chatchat experiment, "text2vec-base-multilingual" was utilized as the word embedding tool. To ensure consistency, GPT-3.5-turbo was standardized as the generator across all experiments.

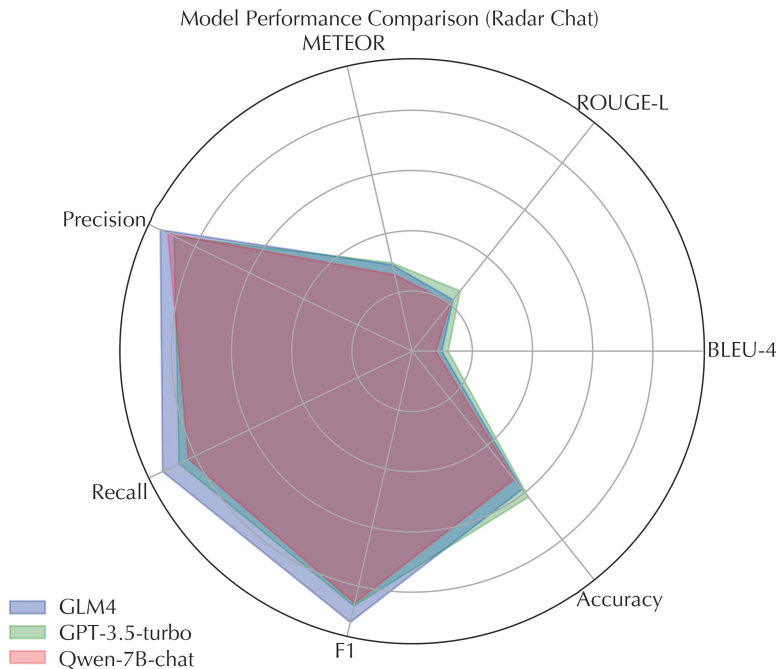


Figure 2. Radar Chart of the Performance of Different Models.

As shown in Table 6, BHT-RAG exhibited relatively superior performance across groups. It achieved the highest BLEU and METEOR scores in Groups 1 and 3, validating its excellent performance in retrieval and generation. TF-IDF and BM25 exhibited stable performance across groups, yet their scores were generally unremarkable. LangChain-Chatchat performed exceptionally in Group 1, with a METEOR score of 27.6593, underscoring its efficacy in data generation for this group. However, LangChain-Chatchat underperformed against BHT-RAG in Groups 2 and 3, particularly in BLEU and ROUGE scores, while RAPTOR secured the highest scores in Group 2.

Table 6. Performance of Different Models in the Question-Answering Task.

Model	Group 1			Group 2			Group 3		
	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR
TF-IDF	16.2299	36.6806	39.7765	9.2298	23.5752	22.9672	7.3780	21.1776	19.5925
BM25	14.2270	30.2082	32.2911	9.3100	22.2920	20.5597	6.7875	19.8200	18.4247
LangChain-chatchat	13.5200	28.6111	34.3840	9.8900	22.9865	27.6593	7.0982	18.6002	24.0886
RAPTOR	15.3366	30.7758	40.1721	11.9769	25.4185	27.9375	8.5302	21.3892	22.6729
BHT-RAG	16.7366	31.1675	40.8344	11.9700	24.3400	26.7400	8.7186	21.6494	23.5965

Table 7 presents the performance of different RAG methods in the multiple-choice task, including comparisons of precision, recall, and F1 scores. Evidently, BHT-RAG continued to demonstrate outstanding performance, particularly achieving the highest precision and F1 scores in Group 1. In Groups 2 and 3, BHT-RAG also performed relatively well, ranking the highest with F1 scores of 87.32% and 83.80%, respectively.

Table 7. Performance of Different Models in the Multiple-Choice Task.

Model	Group 1			Group 2			Group 3		
	Precision/%	Recall/%	F1/%	Precision/%	Recall/%	F1/%	Precision/%	Recall/%	F1/%
TF-IDF	87.98	86.64	87.23	83.42	82.81	83.06	85.04	84.11	84.47
BM25	81.33	81.97	81.38	79.38	78.84	78.89	81.11	77.27	79.06
LangChain-chatchat	75.66	74.67	74.67	72.51	70.93	70.99	73.70	72.57	72.82
RAPTOR	89.75	87.60	88.57	86.76	83.82	85.19	82.40	80.95	81.62
BHT-RAG	90.19	89.67	89.93	87.93	86.89	87.32	85.39	82.71	83.80

In contrast, TF-IDF demonstrated robust performance in Groups 2 and 3, particularly in Group 3, where its recall was comparable to that of BHT-RAG, indicating a robust recall capability in the multiple-choice task. The performance of BM25 was somewhat inferior to that of TF-IDF and BHT-RAG, particularly in Group 3, where all its metrics were relatively low, indicating deficiencies in its retrieval effectiveness. The performance of LangChain-Chatchat was comparatively weak, with precision and F1 scores lower than other models, indicating its challenges with multiple-choice tasks. However, it showed consistent recall performance in some groups, notably with a 70.93% rate in Group 2. RAPTOR's performance across the three groups was only surpassed by BHT-RAG, indicating significant strengths, yet it remained marginally behind the model under study.

Table 8 presents the performance of different RAG models in the fill-in-the-blank task, specifically measuring scores across three different groups. In contrast to multiple-choice questions, fill-in-the-blank tasks typically entailed open-ended responses, necessitating that models not only accurately retrieved pertinent information but also demonstrated robust generation and language comprehension abilities. Consequently, this task was more appropriate for evaluating a model's ability to generate information and maintain accuracy in complex contexts.

Obviously, BHT-RAG outperformed all groups, with a notably high score of 80.2% in Group 1, markedly higher than other models. This result suggested that BHT-RAG's hierarchical tree structure was well-adapted to complex retrieval tasks, effectively clustering relevant information to produce more accurate responses. Furthermore, LangChain-Chatchat demonstrated robust performance in the

Table 8. Accuracy of Different Models in the Fill-in-the-Blank Task

Model	Group		
	1	2	3
TF-IDF	37.62	52.48	49.50
BM25	33.66	34.65	27.72
LangChain-chatchat	66.34	49.5	47.52
RAPTOR	77.23	60.40	52.48
BHT-RAG	80.20	63.37	50.49

fill-in-the-blank task, achieving a high score of 66.34% in Group 1, indicating its capability for effective generation under specific conditions. However, in comparison to BHT-RAG, LangChain-Chatchat still faced limitations regarding information accuracy and comprehensiveness. Moreover, TF-IDF and BM25 showed relatively weak performance, with BM25 scoring substantially lower than other models in all three groups. As BM25 and TF-IDF were primarily based on keyword matching and simple statistical models, they tended to underperform when faced with tasks requiring context understanding and complex reasoning. This observation underscored the constraints of conventional retrieval techniques in the context of generative tasks. The RAPTOR model exhibited greater consistency, achieving the highest score in Group 3.

4.3.2. Ablation Experiments

To validate the necessity of the clustering module in the BHT-RAG model, an ablation experiment was conducted without it. Table 9 presents the model performance in three tasks without clustering. Arrows in the table indicate the metric changes relative to the BHT-RAG model, with upward arrows representing performance improvement and downward arrows indicating a decline. “No-cluster” in the table refers to the results without clustering.

As indicated by the data, for the question-answering task, the BLEU-4 and ROUGE-L metrics were slightly higher without clustering, while the METEOR metric remained essentially unchanged. Analysis revealed that clustering and summarizing with a large language model led to the reorganization and synthesis of node content, resulting in the omission of some irrelevant words. This restructuring resulted in a degree of word omission and word order rearrangement in the retrieved text, impacting the BLEU-4 and ROUGE-L metrics. However, the overall sentence semantics remained largely unchanged. The METEOR metric prioritized semantic alignment between sentences, ensuring that core semantic information was preserved despite changes in sentence structure, which accounted for its stability.

Table 9. Comparison of the Performance between the Model without Clustering and BHT-RAG across Different Tasks.

Group	Model	Question-Answering			Multiple-Choice			Fill-in-the-Blank
		BLEU-4	ROUGE-L	METEOR	Precision/%	Recall/%	F1/%	Accuracy/%
Group 1	No-Cluster	17.0382↑	31.6978↑	40.2570↓	87.59↓	87.92↓	87.62↓	52.48↑
	BHT-RAG	16.7366	31.1675	40.8344	90.19	89.67	89.93	50.49
Group 2	No-Cluster	12.6051↑	25.5304↑	27.7748↑	86.17↓	86.64↓	86.33↓	61.39↓
	BHT-RAG	11.9700	24.3400	26.7400	87.93	86.89	87.32	63.37
Group 3	No-Cluster	9.1214↑	21.7637↑	23.1300↓	80.79↓	80.50↓	80.60↓	79.21↓
	BHT-RAG	8.7186	21.6494	23.5965	85.39	82.71	83.80	80.20

Figure 3 illustrates the grouped bar charts. The performance of BHT-RAG and the model without clustering was compared across the three tasks. In the multiple-choice and fill-in-the-blank tasks, the control group without clustering demonstrated a decline in all three metrics. This was attributed to the fact that, during the text segmentation process, originally coherent paragraphs were split, resulting in the loss of contextual information and the weakening of the semantic connections between paragraphs. Furthermore, the aggregation of relevant questions in the retrieval process was reduced, and logical coherence was weakened, resulting in similar questions being more dispersed in the retrieval results. In contrast, the clustering operation within the BHT-RAG structure produced a different effect.

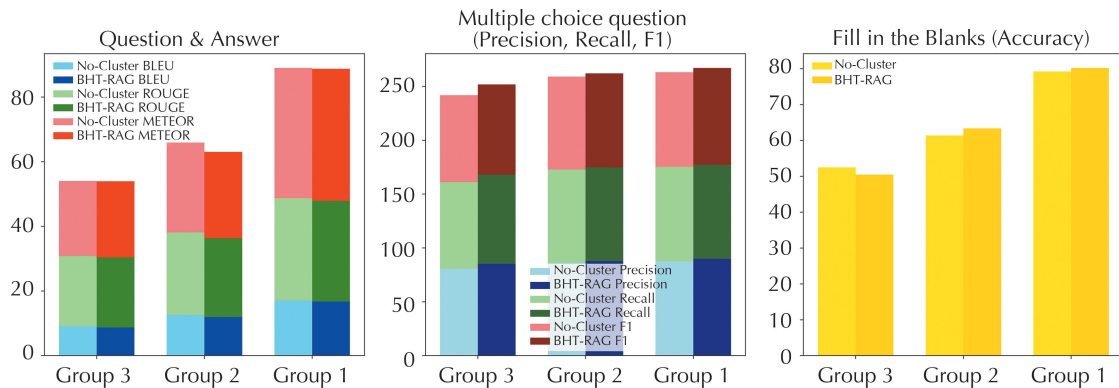


Figure 3. Grouped Bar Chart of the Performance of BHT-RAG and the Model without Clustering across Three Tasks.

4.3.3. Other Public Datasets

To further validate the effectiveness of the model and its capacity for generalization across different datasets, the publicly available datasets Wikieval [53] and L-eval [54] were hereby selected for experimentation. Given the constraints of the datasets and the evaluation tasks employed in this study, the experiments were conducted on a subset of tasks, namely multiple-choice and question-answering tasks. In order to conduct the question-answering task, all 50 samples from the “wikiEval.csv” file in the Wikieval dataset and all 68 question-answer samples from the “financial_qa.jsonl” file in the L-eval dataset were randomly selected. For the multiple-choice task, 100 questions were randomly selected from the “Coursera.jsonl” file in the L-eval dataset. Table 10 illustrates the model’s performance on the question-answering tasks across different datasets, while its performance on the multiple-choice tasks is presented in Table 11.

Table 10. Performance of Different Models on Question-Answering Tasks Across Various Datasets.

Model	MSeval			Wikieval			L-eval		
	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR
TF-IDF	16.2299	36.6806	39.7765	20.4153	28.9492	28.7186	7.9625	18.7439	25.0763
BM25	14.2270	30.2082	32.2911	19.6422	27.7079	27.8190	7.4810	18.3631	23.5565
LangChain-chatchat	13.5200	28.6111	34.3840	18.8570	27.6523	28.9485	5.3421	16.0026	22.1402
RAPTOR	15.3366	30.7758	40.1721	17.0258	25.4058	28.5468	7.0389	17.9113	27.3142
BHT-RAG	16.7366	31.1675	40.8344	19.2159	26.8711	29.5875	7.2171	18.2992	26.7026

Table 11. Performance of Different Models on the L-eval Multiple-Choice Task.

Model	L-eval		
	Precision/%	Recall/%	F1/%
TF-IDF	48.58	39.29	38.48
BM25	50.99	39.45	40.49
LangChain-chatchat	40.16	35.56	35.51
RAPTOR	63.12	45.39	48.82
BHT-RAG	64.07	47.53	50.11

In the question-answering tasks outlined in Table 10, BHT-RAG demonstrated consistent performance across the MS, Wikieval, and L-eval datasets, achieving stable scores on the BLEU, ROUGE, and METEOR

metrics. This performance outstripped that of other models. Notably, the TF-IDF model showed significant strength in the BLEU-4 and ROUGE-L metrics for question-answering tasks, indicating that the lexical and syntactic structures of the retrieved answers were more aligned with the reference answers. This alignment was likely due to the fact that clustering and summarization processes in the study's model and others might disrupt the original semantic order, making direct word matching less effective compared to TF-IDF. However, the TF-IDF model demonstrated inferior performance in the METEOR metric relative to BHT-RAG. This occurred because METEOR not only emphasized lexical matching but also accounted for inflections and synonym matching, thereby better capturing the semantic accuracy of the answers.

The experimental results of the L-eval multiple-choice task in Table 11 revealed that the proposed model significantly outperformed other comparative models in terms of precision, recall, and F1 score, achieving a precision of 64.07% and an F1 score of 50.11%, demonstrating strong generalization ability. However, the overall metrics in this task were relatively low. Analysis revealed a significant number of multiple-choice questions, where the model demonstrated instances of missing or incorrect selections, resulting in a decline in overall performance metrics.

In summary, BHT-RAG, with its hierarchical tree-based retrieval and generation method, demonstrated the best overall performance across different groups and datasets, validating its effectiveness and advantages within the RAG framework.

4.4 Error Analysis

In the question-answering task, the performance of the BHT-RAG model on BLEU-4 and ROUGE-L metrics was occasionally comparable to, or even worse than, that of traditional sparse retrieval methods such as TF-IDF. Analysis suggested that this might be due to the disruption of sentence order during clustering and summarization by BHT-RAG. Given the emphasis of BLEU-4 on matching consecutive four-gram sequences and that of ROUGE-L on matching the longest common subsequence, both metrics were sensitive to sentence order. Consequently, the model's performance on these metrics might be negatively affected. In contrast, the METEOR metric evaluated the semantic information of sentences, which was less impacted by changes in sentence order, thereby preserving overall semantic integrity.

In the multiple-choice task, the primary limitation of the BHT-RAG model was the performance of the generator. For instance, in the MSeval dataset question "What is the Groups of Vinblastine?" BHT-RAG retrieved relevant information stating that "it is classified as a type of Small Molecule and falls under the groups Approved." However, the model still incorrectly selected Option A: Vinca alkaloids. This demonstrated deficiencies in the generator's semantic understanding. To address this issue, future work was proposed to employ more advanced generator models, such as GPT-4.

In the L-eval dataset question "Which of the following statements is NOT true about a 'fork' in the context of version control?" the correct answer included:

A. A fork is when multiple people make changes to the same file and Git is unable to merge the edits.

B. A fork is when the same file has two simultaneous copies.

D. A fork is to copy other people’s code on GitHub.

However, the model only identified Option A. Analysis of the retrieved information indicated that the failure to select Options B and D was primarily due to insufficient retrieval coverage and limitations in the generator’s semantic comprehension.

Future research will investigate hybrid retrieval methods integrating BHT-RAG with traditional sparse retrieval techniques such as TF-IDF to bolster multi-channel recall. Furthermore, the exploration of advanced generator models to optimize RAG performance will be undertaken.

4.5 Compare BHT-RAG with GraphRAG

GraphRAG, developed by Microsoft [9], represents the current SOTA in retrieval-augmented generation. It integrates graph-based reasoning with retrieval mechanisms, excelling in capturing complex relationships between entities. The model shows robust performance in structured information retrieval and reasoning tasks, attributed to its effective modeling of interdependencies among retrieved documents.

To better evaluate the strengths and weaknesses of our proposed model, BHT-RAG, in comparison to the current state-of-the-art model GraphRAG, experiments on different tasks across various groups were hereby conducted using both models. For fairness, the same word embedding model, “text-embedding-ada-002,” and generator model, “GPT-3.5-turbo,” were employed for GraphRAG. Additionally, version 0.1.1 of GraphRAG was used. Table 12 presents the evaluation results of both models across tasks and groups.

Table 12. Performance of GraphRAG and BHT-RAG in Different Groups under Different Tasks.

Tasks	Group	Group 1			Group 2			Group 3		
	Metrics	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR
Q&A	GraphRAG	3.6954	12.8670	22.4695	3.0461	13.0265	20.8021	2.3179	11.1869	16.9957
	BHT-RAG	16.7366	31.1675	40.8344	11.97	24.34	26.74	8.7186	21.6494	23.5965
	Metrics	Precision/ %	Recall/ %	F1/ %	Precision/ %	Recall/ %	F1/ %	Precision/ %	Recall/ %	F1/ %
choice	GraphRAG	90.29	86.25	87.87	93.87	89.89	91.48	81.36	72.96	76.73
	BHT-RAG	90.19	89.67	89.93	87.93	86.89	87.32	85.39	82.71	83.8
	Metric	Accuracy/%								
fill-blank	GraphRAG	87.13			73.27			55.45		
	BHT-RAG	80.20			63.37			50.49		

As shown in Table 12, GraphRAG significantly underperformed BHT-RAG in the QA task across all three metrics. In Group 1, GraphRAG achieved its highest scores in the QA task, with BLEU-4, ROUGE-L, and METEOR scores of 3.6954, 12.8670, and 22.4695, respectively, which lagged behind BHT-RAG's scores in the same group by 13.0412, 18.3005, and 18.3649, respectively. Upon analysis, this disparity was attributed to GraphRAG's reliance on structured knowledge representation and its emphasis on reorganizing and summarizing knowledge. This process significantly altered the original sentence order, leading to subpar performance in generating natural language answers and lower scores across all metrics compared to BHT-RAG.

In the choice task, GraphRAG achieved its best performance in Group 2, involving precision, recall, and F1 scores of 93.87%, 89.89%, and 91.48%, respectively. Moreover, GraphRAG outperformed BHT-RAG in precision within Group 1, yet fell short in other metrics across the remaining groups.

In the fill-in-the-blank task, GraphRAG demonstrated a clear advantage over BHT-RAG, highlighting its strength in comprehensive knowledge retrieval and contextual understanding. This could be attributed to GraphRAG's graph-based reasoning, which better captured contextual relationships.

Experiments were also conducted on the Wikieval and L-eval datasets, as shown in Table 13. The results aligned with those from the MSeval dataset for QA tasks, where GraphRAG exhibited poor BLEU-4, ROUGE-L, and METEOR scores. In the choice task, GraphRAG achieved slightly higher recall than BHT-RAG, yet underperformed in precision, leading to an overall F1 score higher than that of BHT-RAG.

Table 13. Performance of GraphRAG and BHT-RAG in Different Datasets.

Tasks	Datasets	Wikieval			L-eval		
	Metrics	BLEU-4	ROUGE-L	METEOR	BLEU-4	ROUGE-L	METEOR
Q&A	GraphRAG	4.5798	10.4531	15.839	3.0282	10.8556	20.0463
	BHT-RAG	19.2159	26.8711	29.5875	7.2171	18.2992	26.7026
choice	Metrics	–	–	–	Precision/%	Recall/%	F1/%
	GraphRAG	–	–	–	56.65	48.77	50.19
	BHT-RAG	–	–	–	60.14	45.17	47.52

Despite its effectiveness in certain scenarios, GraphRAG is still a computationally intensive model. Its graph construction and reasoning processes are resource-heavy, making it less efficient in terms of runtime and memory usage compared to lightweight alternatives.

Throughout a complete retrieval-based QA process, GraphRAG requires exhaustive knowledge extraction, element summaries, graph summaries, and community summaries. In the indexing phase, it involves extracting entities, relationships, and statements from raw text, performing community detection

within entities, generating multi-granularity community summaries and reports, and ultimately answering prompts based on the document. This process necessitates extensive LLM calls, particularly during knowledge graph construction, which demands comprehensive knowledge extraction and consumes substantial computational resources. Table 14 presents statistics on the stages requiring LLM usage for both the GraphRAG and BHT-RAG models. It was observed that the model in this paper required merely LLM calls during cluster summarization and final answer generation, with no need for LLM in other processes.

Table 14. LLM-Using Stages in GraphRAG and BHT-RAG.

Model	Step						
	Knowledge Extraction	Elements Summaries	Graph Summaries	Community Summaries	Query knowledge Extraction	Generate Community Summaries	Ultimately Answering
GraphRAG	√	√	√	√	√	√	√
BHT-RAG				√			√

5. RAG-ASSISTED PROTEIN IDENTIFICATION

Protein identification from mass spectrometry (MS) data involves matching experimental results with protein information in reference databases. This process includes key steps such as data preprocessing, database searching and matching, and result validation. Mass spectrometry data typically contain essential elements such as mass-to-charge ratios (m/z), peptide sequence features, primary MS data, and secondary MS data, which collectively forge a solid foundation for protein identification.

In the preprocessing stage, key features were extracted from the experimental data, such as mass-to-charge ratios and fragment ion patterns from secondary spectra, serving as preliminary filters for matching against proteins or peptides in the database. The BHT-RAG model could assist in this phase by rapidly retrieving relevant literature or database information, thereby providing annotation support for complex experimental data, such as identifying common modifications or peptide characteristics under specific experimental conditions. Additionally, MS analysis software was employed to interpret the data and generate an initial list of peptide candidates. Enzyme digestion rules, as dictated by the experimental design, were subsequently employed to exclude non-compliant peptides from further analysis.

The subsequent search and matching step began with comparing the mass-to-charge ratios from primary MS data against theoretical values for peptides in the reference database. A certain tolerance was allowed to account for experimental mass deviations. The initial filtering substantially reduced the candidate peptide pool, thereby enhancing computational efficiency. Subsequent use of secondary MS fragment ion features refined the matches through spectral similarity score calculations, identifying the most probable peptide sequences. The BHT-RAG model augmented this process by associating functional and biological

context with candidate peptides, integrating disease-related data or metabolic pathway information to produce a meaningful, high-quality list of candidate proteins.

The final step involved statistical validation of each candidate protein based on MS matching scores, such as posterior error probability (PEP) and false discovery rate (FDR) control. These scores were utilized to rank candidates and determine their prioritization. Additional validation steps, such as calculating peptide coverage or conducting experimental replication, should necessarily be implemented to ensure the accuracy and reliability of the identified proteins. During this validation phase, the BHT-RAG model could retrieve literature, functional annotations, and biological background information regarding candidate proteins, automatically generating validation reports and facilitating to uncover key biological mechanisms or potential applications.

For example, given a peptide with a primary MS mass-to-charge ratio of 1000.4351 m/z and secondary MS fragment patterns (b-ion and y-ion series), these parameters served as critical initial screening criteria. Matching this peptide against all potential candidates in the database, followed by integrating its functional annotations and experimental conditions, allowed the BHT-RAG model to retrieve relevant biological context and functional mechanisms. This resulted in a biologically significant list of candidate proteins. Ultimately, these results could support downstream research, including biomarker discovery, drug target analysis, or therapeutic strategy optimization, offering essential insights into further studies.

6. CONCLUSION

The application of RAG technology in mass spectrometry data improves the retrieval accuracy, simplifies the data analysis, supports knowledge discovery, enhances the Q&A capability, optimizes diagnostic prediction, reduces the noise interference, and constructs a knowledge base, etc., thereby significantly promoting the utilization and application construction of mass spectrometry data. The rational and effective use of RAG in mass spectrometry establishes a novel framework for mass spectrometry data analysis and offers more possibilities for the in-depth application of mass spectrometry technology in medicine and biology.

This paper proposed and implemented a novel retrieval-augmented method, known as Birch-based hierarchical tree retrieval-augmented generation method (BHT-RAG). Through Birch-based hierarchical clustering of documents, this method iteratively generated and selected the most relevant information, considerably improving retrieval and generation performance in complex tasks. To address the gap in mass spectrometry data evaluation for RAG, a specialized medical mass spectrometry RAG dataset was constructed, involving retrievable documents, question-answer pairs, multiple-choice questions, and fill-in-the-blank questions. Upon the construction of the dataset, various combinations of clustering methods and large language models were tested to evaluate their generalization capabilities across multiple tasks. Following comprehensive consideration, GPT-3.5-turbo combined with Birch clustering was hereby identified as the best-performing combination. Subsequently, under identical conditions with a unified

generator, different retrievers were compared. By maintaining GPT-3.5-turbo as the generator, experiments were conducted with retrievers such as BM25 and others.

The superiority of BHT-RAG was validated across question-answer pairs, multiple-choice questions, and fill-in-the-blank tasks, especially in the latter two, where BHT-RAG demonstrated exceptional performance. To further explore the necessity of the clustering module within BHT-RAG, an ablation experiment was conducted on a version of the model without the clustering module, further confirming the module's importance for information integration in BHT-RAG. Additionally, two publicly available datasets were utilized to comprehensively assess the model effectiveness. The results confirmed the superior performance of BHT-RAG across different tasks, effectively leveraging the multi-level information structure to balance retrieval efficiency and generation quality.

Despite BHT-RAG's outstanding performance in multiple tasks, the model is still subjected to certain limitations. First, the model does not provide an in-depth analysis of the necessity of each module in evaluating its overall capability. For example, the performance without the clustering operation and the effects of using other tokenization tools have not been thoroughly tested. Additionally, the model may lose critical information during re-clustering and summarization, potentially impacting its generalization ability. Moreover, the model's generator exhibits deficiencies in semantic understanding, leading to errors in tasks such as multiple-choice questions. The retrieval process also suffers from inadequate information coverage, further limiting its accuracy. These issues necessitate further research and experimental validation.

Future work should focus on the following aspects:

1. Hybrid retrieval methods. To address the limitations in information coverage, further efforts should be made to integrate BHT-RAG with traditional sparse retrieval methods such as TF-IDF, leveraging multi-channel recall to enhance retrieval effectiveness.
2. Advanced generator models. To overcome deficiencies in semantic understanding, more sophisticated generator models, such as GPT-4, should be explored to improve BHT-RAG's performance across complex tasks.
3. Application scenarios in mass spectrometry. Building on the distinctive characteristics of mass spectrometry data, the RAG model's effectiveness in knowledge discovery should be consistently improved to broaden the role of mass spectrometry in scientific research and clinical applications, thereby supporting a wider range of use cases with more robust data-driven insights.
4. Module optimization. Subsequent efforts will investigate methods to mitigate information loss and bolster model generalization. A systematic examination of the interplay between modules will elucidate the performance implications of various configurations, facilitating the advancement towards more refined and adaptable model frameworks.

5. Broadened retrieval scope. BHT-RAG was primarily applied to text tasks in this experiment. Future research should expand its application to cross-modal tasks, such as image-text generation or multimodal information retrieval to increase its versatility, thereby facilitating its broader application in diverse real-world scenarios.

AUTHOR CONTRIBUTIONS

Feilong Wang: Conceptualization, Writing – original draft; Xinyi Cui: Data curation; Formal analysis; Donghui Shi: Supervision, Funding acquisition.

ACKNOWLEDGEMENTS

This work was supported by Anhui Provincial Joint Key Laboratory for Mass Spectrometry Key Technology R&D and Clinical Application (Grant No. 2023ZPLH07); Education Teaching and Research Project of China Association of Construction Education (Grant No. 2023003); Humanity and Social Science Research Project of Anhui Provincial Education Department (Grant No. 2022AH050224); Director's Fund of the Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving, Anhui Jianzhu University (Grant No. IBES2024ZR01).

REFERENCES

- [1] G. Chen, C. Wang, and S. Li, "Research on the effect of CO₂ and H₂O on NO reduction of biomass char by the equivalent characteristic spectrum method via an on-line mass spectrometer," *Energy*, vol. 312, Art. no. 133438, Dec. 2024, doi: 10.1016/j.energy.2024.133438.
- [2] M. Xu, J. Li, Y. Bi, J. Song, and C. Fu, "A method for evaluating the initial addition of tert-butylhydroquinone to oils during storage condition based on gas chromatography-mass spectrometry," *J. Food Compos. Anal.*, vol. 136, Art. no. 106835, Dec. 2024, doi: 10.1016/j.jfca.2024.106835.
- [3] L. B. Carvalho *et al.*, "Normalization methods in mass spectrometry-based analytical proteomics: A case study based on renal cell carcinoma datasets," *Talanta*, vol. 266, Art. no. 124953, Jan. 2024, doi: 10.1016/j.talanta.2023.124953.
- [4] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html>
- [5] X. Li, "Application of RAG model based on retrieval enhanced generation technique in complex query processing," *Advances in Computer, Signals and Systems*, vol. 8, no. 6, pp. 47–53, 2024, doi: 10.23977/acss.2024.080608.
- [6] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," arXiv:2312.10997, 2023. [Online]. Available: <https://arxiv.org/abs/2312.10997>
- [7] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "RAPTOR: Recursive abstractive processing for tree-organized retrieval," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/hash/8a2acd174940dbca361a6398a4f9df91-Abstract-Conference.html

- [8] S. Akesson and F. A. Santos, "Clustered Retrieved Augmented Generation (CRAG)," arXiv:2406.00029, 2024. [Online]. Available: <https://arxiv.org/abs/2406.00029>
- [9] D. Edge et al., "From local to global: A graph RAG approach to query-focused summarization," arXiv:2404.16130, 2024. [Online]. Available: <https://arxiv.org/abs/2404.16130>
- [10] G. S. Ochoa, P. E. Sudol, T. J. Trinklein, and R. E. Synovec, "Class comparison enabled mass spectrum purification for comprehensive two-dimensional gas chromatography with time-of-flight mass spectrometry," *Talanta*, vol. 236, Art. no. 122844, Jan. 2022, doi: 10.1016/j.talanta.2021.122844.
- [11] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, PMLR, vol. 119, 2020, pp. 3929–3938. [Online]. Available: <https://proceedings.mlr.press/v119/guu20a.html>
- [12] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24824–24837, doi: 10.52202/068431-1800.
- [13] A. Zeng et al., "GLM-130B: An open bilingual pre-trained model," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: <https://mlanthology.org/iclr/2023/zeng2023iclr-glm130b/>
- [14] Z. Wan et al., "GPT-RE: In-context learning for relation extraction using large language models," in *Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, 2023, pp. 3534–3547, doi: 10.18653/v1/2023.emnlp-main.214.
- [15] J. Wei et al., "Emergent abilities of large language models," *Trans. Mach. Learn. Res.*, 2022. [Online]. Available: <https://mlanthology.org/tmlr/2022/wei2022tmlr-emergent/>
- [16] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf
- [17] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: <https://www.jmlr.org/papers/v21/20-074.html>
- [18] T. B. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/hash/1457c0d6bfc4967418bfb8ac142f64a-Abstract.html
- [19] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald, "On faithfulness and factuality in abstractive summarization," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2020, pp. 1906–1919, doi: 10.18653/v1/2020.acl-main.173.
- [20] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proc. 16th Conf. European Chapter Assoc. Comput. Linguistics (EACL)*, 2021, pp. 874–880, doi: 10.18653/v1/2021.eacl-main.74.
- [21] F. Petroni et al., "Language models as knowledge bases?," in *Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 2463–2473, doi: 10.18653/v1/D19-1250.
- [22] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in *Proc. 2020 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781, doi: 10.18653/v1/2020.emnlp-main.550.
- [23] K. Kreimeyer, J. V. Canzoniero, M. Fatteh, V. Anagnostou, and T. Botsis, "Using retrieval-augmented generation to capture molecularly-driven treatment relationships for precision oncology," *Stud. Health Technol. Inform.*, vol. 316, pp. 983–987, 2024, doi: 10.3233/SHTI240575.

- [24] J. Li, Y. Lei, Y. Bian, D. Cheng, Z. Ding, and C. Jiang, "RA-CFGPT: Chinese financial assistant with retrieval-augmented large language model," *Front. Comput. Sci.*, vol. 18, no. 5, Art. no. 185350, 2024, doi: 10.1007/s11704-024-31018-5.
- [25] Y. Sha, Y. Feng, M. He, S. Liu, and Y. Ji, "Retrieval-augmented knowledge graph reasoning for commonsense question answering," *Mathematics*, vol. 11, no. 15, Art. no. 3269, 2023, doi: 10.3390/math11153269.
- [26] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting for retrieval-augmented large language models," in *Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, 2023, pp. 5303–5315, doi: 10.18653/v1/2023.emnlp-main.322.
- [27] W. Peng et al., "Large language model based long-tail query rewriting in Taobao search," in *Companion Proc. ACM Web Conf. 2024*, Singapore, 2024, pp. 20–28, doi: 10.1145/3589335.3648298.
- [28] H. S. Zheng et al., "Take a step back: Evoking reasoning via abstraction in large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/hash/592da1445a51e54a3987958b5831948f-Abstract-Conference.html
- [29] L. Gao, X. Ma, J. Lin, and J. Callan, "Precise zero-shot dense retrieval without relevance labels," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2023, pp. 1762–1777, doi: 10.18653/v1/2023.acl-long.99.
- [30] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," in *Findings Assoc. Comput. Linguistics: EMNLP 2023*, 2023, pp. 9248–9274, doi: 10.18653/v1/2023.findings-emnlp.620.
- [31] W. Yu et al., "Generate rather than retrieve: Large language models are strong context generators," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: <https://www.microsoft.com/en-us/research/publication/generate-rather-than-retrieve-large-language-models-are-strong-context-generators/>
- [32] X. Cheng, D. Luo, X. Chen, L. Liu, D. Zhao, and R. Yan, "Lift yourself up: Retrieval-augmented text generation with self-memory," in *Advances in Neural Information Processing Systems*, vol. 36, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/hash/887262aeb3eafb01ef0fd0e3a87a8831-Abstract.html
- [33] S. Wang et al., "Training data is more valuable than you think: A simple and effective method by retrieving from training data," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2022, pp. 3170–3179, doi: 10.18653/v1/2022.acl-long.226.
- [34] D. Cheng et al., "UPRISE: Universal prompt retrieval for improving zero-shot evaluation," in *Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, 2023, pp. 12318–12337, doi: 10.18653/v1/2023.emnlp-main.758.
- [35] Z. Dai et al., "Promptagator: Few-shot dense retrieval from 8 examples," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: <https://mlanthology.org/iclr/2023/dai2023iclr-promptagator/>
- [36] O. Khattab et al., "Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP," arXiv:2212.14024, 2022. [Online]. Available: <https://arxiv.org/abs/2212.14024>
- [37] Z. Ke, W. Kong, C. Li, M. Zhang, Q. Mei, and M. Bendersky, "Bridging the preference gap between retrievers and LLMs," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2024, pp. 10438–10451, doi: 10.18653/v1/2024.acl-long.562.
- [38] X. V. Lin et al., "RA-DIT: Retrieval-augmented dual instruction tuning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024. [Online]. Available: https://proceedings.iclr.cc/paper_files/paper/2024/hash/536d18fbb454f80221465f1a42c6f389-Abstract-Conference.html

- [39] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, "Okapi at TREC-3," in *Proc. 3rd Text REtrieval Conf. (TREC-3)*, Gaithersburg, MD, USA, 1994, NIST Special Publication 500-225, pp. 109-126, doi: 10.6028/NIST.SP.500-225.routing-city.
- [40] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Doc.*, vol. 28, no. 1, pp. 11-21, 1972, doi: 10.1108/EB026526.
- [41] A. Salemi, J. A. Pizzorno, and H. Zamani, "A symmetric dual encoding dense retrieval framework for knowledge-intensive visual question answering," in *Proc. 46th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Taipei, Taiwan, 2023, pp. 110-120, doi: 10.1145/3539618.3591629.
- [42] R. Zhou, "Advanced embedding techniques in multimodal retrieval augmented generation: A comprehensive study on cross modal AI applications," *J. Comput. Electron. Inf. Manag.*, vol. 13, no. 3, pp. 16-22, 2024, doi: 10.54097/h8wf8vah.
- [43] T. Lan, D. Cai, Y. Wang, H. Huang, and X.-L. Mao, "Copy is all you need," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: <https://arxiv.org/abs/2307.06962>
- [44] X. Du and H. Ji, "Retrieval-augmented generative question answering for event argument extraction," in *Proc. 2022 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2022, pp. 4649-4666, doi: 10.18653/v1/2022.emnlp-main.307.
- [45] Z. Guo, S. Cheng, Y. Wang, P. Li, and Y. Liu, "Prompt-guided retrieval augmentation for non-knowledge-intensive tasks," in *Findings Assoc. Comput. Linguistics: ACL 2023*, 2023, pp. 10896-10912, doi: 10.18653/v1/2023.findings-acl.693.
- [46] G. Kim, S. Kim, B. Jeon, J. Park, and J. Kang, "Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models," in *Proc. 2023 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2023, pp. 996-1009, doi: 10.18653/v1/2023.emnlp-main.63.
- [47] Z. Luo *et al.*, "Augmented large language models with parametric knowledge guiding," arXiv:2305.04757, 2023. [Online]. Available: <https://arxiv.org/abs/2305.04757>
- [48] L. Zha *et al.*, "TableGPT: Towards unifying tables, nature language and commands into one GPT," arXiv:2307.08674, 2023. [Online]. Available: <https://arxiv.org/abs/2307.08674>
- [49] T. Chen *et al.*, "Dense X retrieval: What retrieval granularity should we use?," in *Proc. 2024 Conf. Empirical Methods in Natural Language Processing (EMNLP)*, Miami, FL, USA, 2024, pp. 15159-15177, doi: 10.18653/v1/2024.emnlp-main.845.
- [50] Y. Wang, N. Lipka, R. A. Rossi, A. F. Siu, R. Zhang, and T. Derr, "Knowledge graph prompting for multi-document question answering," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 17, pp. 19206-19214, 2024, doi: 10.1609/aaai.v38i17.29889.
- [51] D. Zhou *et al.*, "Least-to-most prompting enables complex reasoning in large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023. [Online]. Available: <https://mlanthology.org/iclr/2023/zhoul2023iclr-leasttomost/>
- [52] S. Dhuliawala *et al.*, "Chain-of-verification reduces hallucination in large language models," in *Findings Assoc. Comput. Linguistics: ACL 2024*, Bangkok, Thailand, 2024, pp. 3563-3578, doi: 10.18653/v1/2024.findings-acl.212.
- [53] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "RAGAs: Automated evaluation of retrieval augmented generation," in *Proc. 18th Conf. European Chapter Assoc. Comput. Linguistics: System Demonstrations*, St. Julian's, Malta, 2024, pp. 150-158, doi: 10.18653/v1/2024.eacl-demo.16.
- [54] C. An *et al.*, "L-Eval: Instituting standardized evaluation for long context language models," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Bangkok, Thailand, 2024, pp. 14388-14411, doi: 10.18653/v1/2024.acl-long.776.

AUTHOR BIOGRAPHY



Feilong Wang is a graduate student. His research interests mainly focus on large language model applications, retrieval-augmented generation (RAG), mass spectrometry data analysis, and natural language processing. (wfl0325@stu.ahjzu.edu.cn): conducted the experiments; methodology; writing original draft.
ORCID: 0009-0008-6161-2525



Donghui Shi a professor in the School of Electronics and Information Engineering at Anhui Jianzhu University. His research interests include Knowledge Graph, Semantic Network, Artificial intelligence. (shidonghui@ahjzu.edu.cn): funding acquisition; writing review and editing; supervision.
ORCID: 0000-0001-5301-368X



Xinyi Cui is a graduate student. Her research interests mainly focus on large language model applications, Knowledge Graph and natural language processing. (cxy0302@stu.ahjzu.edu.cn): data curation; writing-review and editing.
ORCID: 0009-0007-6614-3054