



Article

Research on Sem-RAG: A Corn Planting Knowledge Question-Answering Algorithm Based on Fine-Grained Semantic Information Retrieval Enhancement

Bing Bai ^{1,2,3}, **Xiaoyan Meng** ^{1,2,3,*} and **Chenzi Zhao** ^{1,2,3} ¹ School of Computer and Information Engineering, Xinjiang Agricultural University, Urumqi 830052, China; 320233405@xjau.edu.cn (B.B.); cassieeez666@gmail.com (C.Z.)² Ministry of Education, Engineering Research Center for Intelligent Agriculture, Urumqi 830052, China³ Xinjiang Agricultural Informatization Engineering Technology Research Center, Urumqi 830052, China

* Correspondence: mxy@xjau.edu.cn

Abstract

Large language models and retrieval-augmented generation (RAG) are widely applied in knowledge question-answering tasks. However, in knowledge-intensive domains such as agriculture, hallucination and insufficient retrieval accuracy remain challenging. To address these issues, we propose Sem-RAG, a corn planting knowledge question-answering algorithm based on fine-grained semantic retrieval enhancement. Unlike standard NaiveRAG, which retrieves only fixed-length text chunks, and GraphRAG, which relies solely on graph node connections, Sem-RAG introduces a dual-store retrieval mechanism. It constructs both a surface semantic store (chunk-level embeddings) and a fine-grained semantic store derived from Leiden-based community summaries. These community summaries do not merely shorten contexts; instead, they provide thematic-level semantic aggregation across document chunks, thereby enhancing semantic coverage and reducing noise. During retrieval, user queries are matched against the surface store to locate relevant chunks and simultaneously linked to corresponding thematic summaries in the fine-grained store, ensuring that both local details and higher-level associations are leveraged. We evaluated Sem-RAG on the corn knowledge question-answering dataset CornData. The algorithm achieved Answer-C, Answer-R, and CR scores of 94.6%, 84.6%, and 70.4%, respectively, which were 2.6%, 1.7%, and 1.6% higher than those of traditional NaiveRAG. These results demonstrate that Sem-RAG materially improves the quality and reliability of agricultural knowledge Q&A by combining dual-store retrieval with community-level semantic aggregation.



Academic Editor: George Drosatos

Received: 15 September 2025

Revised: 5 October 2025

Accepted: 6 October 2025

Published: 9 October 2025

Citation: Bai, B.; Meng, X.; Zhao, C. Research on Sem-RAG: A Corn Planting Knowledge Question-Answering Algorithm Based on Fine-Grained Semantic Information Retrieval Enhancement. *Appl. Sci.* **2025**, *15*, 10850. <https://doi.org/10.3390/app151910850>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: large language model; agricultural Q&A; fine-grained semantics; knowledge graph; corn cultivation

1. Introduction

With the increasing demand for precision and intelligent technology in agricultural production, the agricultural sector is facing increasingly complex challenges. As an important grain crop, corn cultivation management involves various aspects of professional knowledge such as variety selection, cultivation techniques, and pest control. The demand for relevant technical support from agricultural practitioners continues to increase. Traditional information acquisition channels, such as books, technical manuals, Internet searches, etc., are widely used, but in the rapidly changing and specialized agricultural production environment, there are often problems such as low efficiency of information

acquisition, lack of professionalism in content, and a lack of personalized services, which make it difficult to meet actual needs.

In recent years, large language models (LLMs) have made significant breakthroughs in the field of Natural Language Processing (NLP). Initially, BERT introduced a bidirectional Transformer architecture [1], which effectively improved the model's ability to understand semantics [2]. Subsequently, the GPT series significantly optimized the coherence and creativity of text generation by adopting an autoregressive generation mechanism [3]. Afterwards, the T5 model further innovatively formulated multiple natural language tasks as a “text to text” conversion problem, thereby enhancing the model's task generalization ability [4]. On this basis, large-scale models such as PaLM have significantly expanded their parameter scale, further improving their performance in context understanding and complex reasoning tasks [5]. Researchers have begun to explore the application of LLMs in vertical fields: Yang et al. [6] proposed a medical question-answering (MedQ&A) method that utilized the inherent medical knowledge and reasoning ability of the Llama large language model to improve the classification performance for medical question-answering under zero-sample learning, providing a new method and practical foundation for the application of LLMs in vertical fields. Md. Salim et al. [7] developed a web application called LLM Q&A Builder, which integrates data preparation and model fine-tuning development, making it easy for both technical and non-technical users to build internal Q&A systems for organizations, providing methodological and tool support for LLMs in different fields such as enterprise information retrieval.

Although LLMs' language comprehension and generation capabilities provide new opportunities for agricultural technology support, there are still limitations in their application in the agricultural field. On the one hand, LLMs rely on training data inference, and closed knowledge systems make it difficult to ensure the real-time and authoritative nature of the information, resulting in problems such as lagging updates and incomplete information; on the other hand, general LLMs lack professional knowledge training in the field of agriculture and are prone to “hallucinations” [8], among which factual errors, hollow content, and logical confusion are the most typical problems, limiting their practicality in agricultural intelligent question answering.

To overcome the problems of knowledge update lag and an insufficient professional Q&A ability in LLMs, retrieval-augmented generation (RAG) [9,10] methods have been widely applied. RAG combines information retrieval and content generation technologies to achieve fast and efficient retrieval from massive amounts of information [11,12]. Importantly, RAG has been increasingly recognized as a fundamental baseline to beat in knowledge-intensive tasks [13]. The core of this technology is to retrieve the text chunks that are closest to the query from the relevant document knowledge base based on user queries and then integrate this text chunk information with input prompt words (Prompt) [14] into context and input it into the large language model as the generation module, thereby enhancing the ability to reference the latest and authoritative knowledge and improving the professionalism and credibility of the answers [15]. At present, this method has been validated for effectiveness in knowledge-intensive fields such as healthcare, agriculture, cybersecurity, and food. Zakka et al. [16] developed the Almanac framework and applied RAG technology to clinical decision support. This framework significantly improves the accuracy and completeness of generated answers in clinical settings by integrating medical guidelines. However, the framework still has limitations in handling queries that cannot directly extract answers from the guide, which suggests the need for cautious deployment and risk mitigation measures in practical applications. Malali et al. [17] studied the application of RAG in financial document processing, proposed using RAG to automate compliance and regulatory reporting processes, and verified its advantages in improving

data accuracy and decision quality. Research has pointed out that the shortcomings of RAG include incomplete accuracy of the results, dependence on external data quality, and insufficient transparency in the retrieval and generation process. Su et al. [18] proposed parameterized RAG, which integrates external knowledge into LLMs through document parameterization to address the limitations of contextual knowledge enhancement methods. Experimental results show that parameterized RAG significantly improves the effectiveness of LLM knowledge enhancement, but the parameterization process is computationally expensive, and the parameterized representation of each document far exceeds the pure text, limiting its scalability; Ding et al. [19] proposed the RealGen framework, which solves the problem of traditional methods finding it difficult to generate unseen scenes by retrieving existing traffic scene examples and generating new scenes based on these contexts. However, the limitation of this framework is that the retrieved examples and generative models may not fully capture complex behavioral contexts, and the input features retrieved and generated are insufficient; Wang et al. [20] proposed MADAM-RAG, a multi-turn question-answering framework based on retrieval-augmented generation (RAG), which combines retrieved multi-document evidence with LLMs and introduces self-reflection and critical evaluation mechanisms to achieve robust handling of potential misleading information. This method ensures accuracy in generating answers, but its generation performance is highly dependent on the quality and completeness of the retrieved documents. When there is misleading information or an uneven distribution in the documents, the performance of the model may be affected; Patrice Béchard [21] proposed a system based on RAG to generate structured outputs of enterprise workflows from natural language requirements, addressing the issue of “hallucinations” in generative artificial intelligence. While combining external retrieval information to reduce hallucinations and improve the generalization ability of LLMs, there are still limitations such as incomplete elimination of hallucinations, dependence on post-processing, and insufficient collaboration between retrievers and LLMs; Shao et al. [22] proposed the ITER-RETEGEN method to enhance retrieval LLMs, using the model’s initial response to task inputs as context-guided retrieval to obtain more relevant knowledge to improve the next round of generation results. This method improves the performance of retrieval generation but still has limitations in prompt optimization and not covering long-text generation tasks; Ovadia et al. [23] compared the effectiveness of unsupervised fine-tuning and RAG in knowledge-intensive tasks and found that RAG outperforms unsupervised fine-tuning in handling both existing and new knowledge during training. They also pointed out that LLMs face difficulties in learning new factual information through unsupervised fine-tuning, and exposing multiple variants of the same fact can alleviate this problem, thus demonstrating the advantages of RAG in expanding and updating LLM knowledge; Lameck et al. [24] conducted a systematic review of the application of RAG-based large language models in the medical field, dividing RAG methods into three paradigms, naive, advanced, and modular, and summarizing the evaluation frameworks and indicators. This review provides a reference for understanding the advantages, limitations, and research gaps of RAG in vertical fields.

Despite the broad prospects of RAG technology, it still faces many challenges. Although the retrieval module of RAG greatly improves the efficiency and accuracy of model retrieval, its effectiveness still needs to be improved when facing fuzzy queries and niche knowledge domains [25]. After obtaining the search results, it is also necessary to consider whether the length of the text can smoothly enter the generator. After inputting the search results into the query generator, it is necessary to think about how to integrate the information into the answer, so as to align the searcher with the generator. In addition, the computational cost of the model used in RAG technology also needs to be considered, and so on. Specifically, in the field of corn cultivation in agriculture, the scale of knowledge

information used for retrieval is large, and the semantic connections between contextual information are relatively close. In line with recent advances in graph-oriented medical RAG that leverage structured, community-aware designs to mitigate hallucinations [26], our approach further emphasizes community-level knowledge structuring to enhance reliability and reduce the risk of misleading responses. Traditional RAG retrieval methods find it difficult to fully express the hierarchical and logical relationships of agricultural knowledge, which limits the knowledge reasoning ability of large language models in agricultural question-answering.

In this study, we aim to address the following research question: How can we enhance the retrieval accuracy and reduce hallucinations in large-language-model-based question-answering systems for knowledge-intensive agricultural domains, such as corn cultivation? Therefore, in order to extract rich semantic information from external knowledge documents as much as possible for corn planting knowledge question-answering tasks in the agricultural field, we propose Sem-RAG, a novel dual-store semantic retrieval framework that combines chunk-level embeddings and community-level thematic summaries to improve both the semantic coverage and answer reliability.

Distinct from standard NaiveRAG, which only retrieves fixed-length text chunks, and GraphRAG, which mainly relies on graph connections, Sem-RAG introduces a dual-store retrieval design. It builds both a surface semantic store that preserves chunk-level embeddings and a fine-grained semantic store constructed from Leiden-based community summaries. The latter not only compresses text but also provides thematic-level semantic aggregation across chunks, which substantially improves answer quality by enhancing knowledge associations and reducing noise. By jointly retrieving from both stores, Sem-RAG achieves a balance between local semantic detail and higher-level contextual reasoning.

As shown above, the main contributions of this paper are as follows:

1. We developed a knowledge question-answering method, Sem-RAG, for performing knowledge question-answering tasks in the agricultural domain, specifically for corn planting. This method is based on the idea of fine-grained semantic capture and enhances the performance of large language models in the aspect of knowledge reasoning by combining surface semantics with context-related semantics.
2. We designed a graph topic module, Graph-Content, for capturing and processing context-related semantic association information from knowledge documents. This module captures triple information with semantic associations from knowledge documents and then performs community layering and a thematic community summary on the captured semantic triples according to the Leiden algorithm, which serves as context-related semantics.
3. We independently constructed a knowledge question-answering dataset, CornData, suitable for evaluating the Sem-RAG method and conducted experimental verification on the Sem-RAG method to demonstrate its superior performance.

2. Materials and Methods

2.1. Algorithm Introduction

The algorithmic principle of Sem-RAG is shown in Algorithm 1, and its overall process is further illustrated in Figure 1. The algorithm is divided into two stages, including the construction of a knowledge base and the retrieval inference stage.

In the stage of building the knowledge base, firstly, the professional document T of corn planting knowledge is divided into fixed-length chunks to form $\{C_1, C_2, \dots, C_n\}$; then, each chunk C_i in $\{C_1, C_2, \dots, C_n\}$ is processed sequentially, semantically related triple information Q_i is extracted from C_i , and a graph G_i is established based on Q_i . Next, we use the Leiden algorithm to stratify the graph G_i into community lists $\{G_{i1}, G_{i2}, \dots, G_{ip}\}$ and generate

a themed community summary S_{ij} for each community G_{ij} , which includes <topic, key relationship, condition, step>; then, we vectorize the original text of each chunk C_i to obtain V_i and vectorize the thematic summary S_{ij} in each chunk C_i to obtain U_{ij} ; finally, based on V_i and U_{ij} , we construct the surface semantic vector knowledge base $VS_{shallow}$ and the fine-grained semantic vector knowledge base VS_{fine} for retrieval, respectively.

Algorithm 1 Semantic retrieval-augmented generation.

Require: Input text T (corn-growing docs) and question P

Ensure: Retrieved answers A

- 1: Parse T into fixed-length chunks $\{C_1, C_2, \dots, C_n\}$
 - 2: **for** each chunk C_i in $\{C_1, C_2, \dots, C_n\}$ **do**
 - 3: Extract semantic triples Q_i from C_i , where $Q_i = \{(entity_1, entity_2, relation)\}$
 - 4: Build a per-chunk graph G_i from Q_i
 - 5: Run the Leiden algorithm on G_i to detect communities $\{G_{i1}, G_{i2}, \dots, G_{ip}\}$
 - 6: **for** each community G_{ij} in $\{G_{i1}, \dots, G_{ip}\}$ **do**
 - 7: Summarize G_{ij} into a thematic summary S_{ij} (topic, key relations, conditions, steps)
 - 8: **end for**
 - 9: Vectorize C_i (original text) with a shallow encoder $\rightarrow V_i$
 - 10: Vectorize each S_{ij} with a fine-grained encoder $\rightarrow U_{ij}$
 - 11: **end for**
 - 12: Build two vector stores: $VS_{shallow} = \{(ID(C_i), V_i)\}$ $VS_{fine} = \{((ID(C_i), ID(G_{ij})), U_{ij})\}$
 - 13: Fine-tune the LLM on a small set of domain-specific Q&A pairs
 - 14: Encode P with the shallow encoder $\rightarrow q_s$
 - 15: Calculate similarity between q_s and all V_i in $VS_{shallow}$; sort and select top-k chunks $\{Z_1, Z_2, \dots, Z_k\}$ with IDs $\{ID(Z_1), \dots, ID(Z_k)\}$
 - 16: Collect fine summaries linked to the retrieved chunks: $U^* = \{\cup_{i,j} | ID(C_i) \in \{ID(Z_1), \dots, ID(Z_k)\}\}$
 - 17: Rank U^* by similarity to q_s and select top-r summaries S^*
 - 18: Provide both the retrieved chunk texts $\{Z_1, \dots, Z_k\}$ and selected summaries S^* to the LLM
 - 19: LLM generates answers A based on $\{Z_1, \dots, Z_k\}$ and S^*
 - 20: Output answers A
-

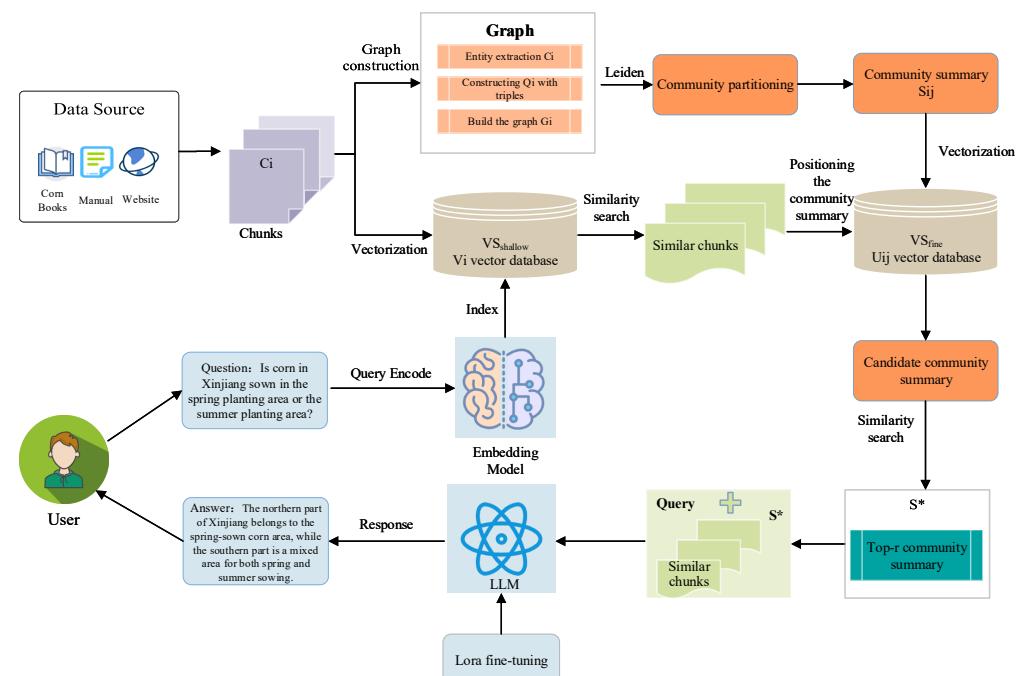


Figure 1. The algorithm principle of Sem-RAG: this algorithm is divided into two stages, including the construction of the knowledge base and the retrieval reasoning stage.

In the retrieval reasoning stage, firstly, LoRA fine-tuning work was carried out on specific domain data for the large language model to enhance its ability to generate replies after retrieval; secondly, the user query vector P was encoded as qs , and we performed similar chunk queries with all Vis on the surface semantic knowledge base $VS_{shallow}$ to obtain a list of similar chunks $\{Z_1, Z_2, \dots, Z_k\}$; then, the candidate graph community summary information in the fine-grained semantic vector knowledge base VS_{fine} is located through the similar chunk list $\{Z_1, Z_2, \dots, Z_k\}$. The user query vector qs is searched for similarity with the candidate graph community summary information to obtain the most relevant r graph community summaries, forming a set of graph summaries S^* . Finally, the similar chunk information $\{Z_1, Z_2, \dots, Z_k\}$ and the graph community summary information S^* are used as semantic information references for the large language model's response to generate query results.

2.2. LoRA Fine-Tuning

LoRA (Low-Rank Adaptation) is a parameter-efficient fine-tuning method [27–29], whose core idea is to introduce trainable “adapters” into the weight matrix of the pre-trained model through low-rank decomposition. Based on freezing the pre-trained model parameters, it achieves efficient adaptation to reduce training and storage costs. In this study, we enhanced the adaptation ability of the large language model to domain knowledge using the LoRA method based on a small number of samples in the field of maize knowledge question-answering. At the same time, the LoRA method was used at the task level to evaluate the extraction capability of graph nodes to align the results of the extraction evaluation.

As shown in Figure 2, the algorithm principle of the LoRA fine-tuning mechanism is to first select the layers that need to be adapted in the large-scale pre-trained model, that is, the attention projection matrix in the Transformer structure. Secondly, in the forward propagation stage, the input vector x is calculated using the weight modification module to obtain the output y . The calculation method of the weight modification module is shown in Equation (1). Among them, the original weight matrix of the attention projection matrix is $W \in R^{d*k}$. The LoRA fine-tuning mechanism does not directly update the original weight matrix but introduces two low-rank matrices $A \in R^{d*r}$ and $B \in R^{r*k}$ and constructs a low-rank update term ΔW through the low-rank update process. Wx is the frozen output of the pre-training part, and $(AB)x$ is the trainable correction provided by the LoRA mechanism. The calculation method for the low-rank update process is shown in Equation (2), where α is the scaling factor used to control the stability of the update amplitude. Then, in the backpropagation stage, only the gradients of low-rank matrices A and B are calculated and updated, while the pre-training parameter W remains unchanged to reduce the number of parameters that need to be updated and stored during the training process; finally, at the end of the training, we save the updated weights ΔW . In the model inference stage, the weight matrix W of the pre-trained model is superimposed with the low-rank updated weights ΔW to obtain the merged weight matrix W' , and model inference is performed based on W' .

The advantages of LoRA include training a very small number of newly added parameters to achieve near full fine-tuning effects; the LoRA weights of different tasks being loaded as modules without affecting the main model; and avoiding catastrophic forgetting. Overall, the core idea of LoRA is to compress the adaptability of large models into a small number of parameter updates through low-rank approximation, thereby achieving efficient transfer learning.

$$y = (W + \Delta W)x \quad (1)$$

$$\Delta W = \frac{\alpha}{r} AB \quad (2)$$

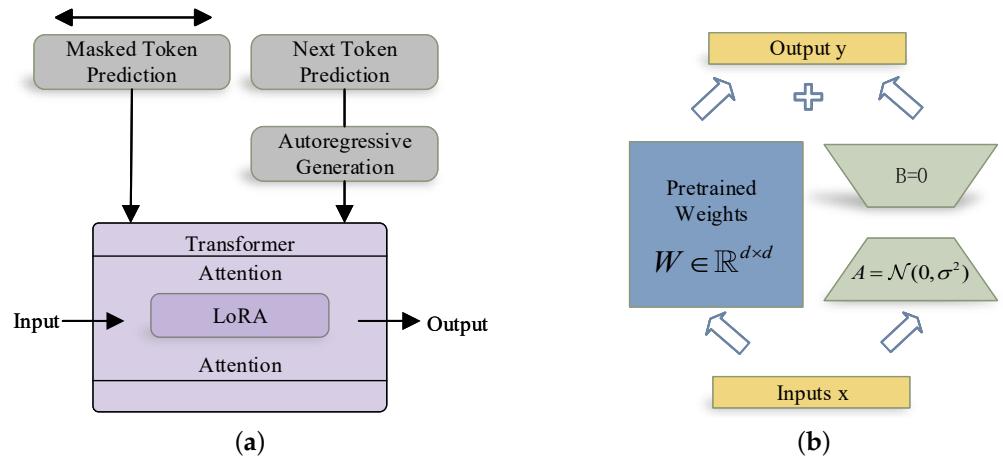


Figure 2. (a) LoRA fine-tuning structure. (b) Algorithm principle of LoRA fine-tuning mechanism.

2.3. Contextual Semantic Association Processing Module

During the process of knowledge base construction, text chunks often contain complex semantic relationships, and it is difficult for direct storage to support efficient semantic retrieval and reasoning. Therefore, this study designs a contextual semantic association processing module, aiming to transform natural language knowledge into structured, thematic, and vectorizable multi-level representations. This module mainly includes steps such as semantic triple information extraction, community division and summary generation, and semantic vectorization.

2.3.1. Triple Extraction and Knowledge Graph Construction

For each text chunk C_i , this study utilizes the LLM's general extraction capability to extract triple information. Unlike traditional dependency syntax analysis and rule-based relationship extraction methods, LLM has stronger generalization and contextual understanding abilities in complex contexts and cross-domain scenarios, thus enabling more accurate recognition of semantic structures.

$$Q_i = \{(h, r, t)\} h, t \in V, r \in R \quad (3)$$

The extracted triples are in the form shown in Equation (3). Here, h represents the head entity, t represents the tail entity, and r represents the semantic relationship. Based on the triple set Q_i , a local knowledge graph can be constructed, and its representation form is as shown in Equation (4). Here, V is the entity set, and E is the set of relationship edges. The representation form of E is as shown in Equation (5).

$$G_i = (V, E) \quad (4)$$

$$E = \{(h, r, t) | (h, r, t) \in Q_i\} \quad (5)$$

2.3.2. Community Partitioning

As the number of triples increases, the scale and complexity of the knowledge graph G_i grow rapidly, necessitating further structural decomposition. This study employs the Leiden community detection algorithm to partition the graph, with the maximization of modularity Q as the optimization objective. The calculation of Q is shown in Equation (6). Here, A_{ij} represents the edge weight between nodes i and j , and k_i denotes the degree of

node i , which is calculated as shown in Equation (7). m is the total weight of the edges in the graph, and δ is the indicator function, taking a value of 1 when nodes i and j belong to the same community.

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) \quad (6)$$

$$k_i = \sum_j A_{ij} \quad (7)$$

The Leiden algorithm introduces three-stage iterative optimization based on the Louvain framework: firstly, during the node movement phase, the community membership of nodes is adjusted to improve local modularity while ensuring community connectivity; secondly, in the community refinement stage, the initial community should be split and optimized to avoid the emergence of disconnected or inferior communities; finally, in the graph aggregation stage, the community is compressed into supernodes, a new graph is constructed, and iterations are performed again until convergence. After Leiden algorithm's community partitioning, a community set with high semantic consistency is ultimately formed.

2.3.3. Community Summary Construction and Semantic Vectorization

In order to improve the readability and retrievability of knowledge within the community, research is conducted on generating thematic semantic summaries S_{ij} for each community G_{ij} . The construction process is as follows: firstly, the keyword extraction process is carried out, and the TextRank algorithm is used to identify words with strong thematic relevance; secondly, relationship filtering is performed, the triples within the community are sorted based on relationship weights, and core knowledge is retained; finally, condition and step extraction is performed, combined with use of the semantic parsing ability of the large language model, to identify conditional patterns and operational processes. The summary S_{ij} formed through this process can be formally represented as (theme, key relationships, conditions, steps). This structured representation compresses complex graph knowledge into interpretable knowledge units, providing support for subsequent fine-grained retrieval.

After obtaining the original chunks and community summaries, they need to be mapped to a vector space to support semantic retrieval. By embedding the thematic community summary S_{ij} through an encoding model, a fine-grained semantic vector U_{ij} is obtained, and finally, a fine-grained semantic knowledge base is constructed. The representation of U_{ij} is shown in Equation (8), where f_θ represents the encoder function. The representation of VS_fine is shown in Equation (9).

$$U_{ij} = f_\theta(S_{ij}) \quad (8)$$

$$VS_{fine} = \{U_{11}, U_{12}, \dots, U_{np}\} \quad (9)$$

3. Experiment and Data Analysis

Research on the performance evaluation of the Sem-RAG algorithm and related fine-tuning of the base model using the self-built corn planting knowledge Q&A dataset Corn-Data was conducted. The dataset description is in Section 3.1, while the experimental environment, evaluation criteria, and detailed results are in Sections 3.2–3.4, respectively.

3.1. Dataset Description

To evaluate Sem-RAG's vertical domain knowledge retrieval and question-answering ability, we collected approximately 300,000 words of corn planting knowledge documents using web crawling technology. Combining OCR technology and large language modeling technology, the corn planting knowledge documents were analyzed and summarized, and finally, 1763 question-answering data points were extracted to form the corn planting knowledge question-answering dataset CornData. Among them, each data point contains fields such as id, question, answer, and content, which, respectively, represent the number, question, correct answer, and specific textual content of the document. A data example for CornData is shown in Table 1.

Table 1. CornData, a dataset for corn planting knowledge Q&A. Each data entry includes fields such as id, question, answer, and content.

Chunk_ID	Question	Answer	Chunk_Content
chunk_1	What are the key points of maize intercropping technology in southern Xinjiang?	The key techniques for intercropping corn in southern Xinjiang include selecting suitable high-yield varieties, controlling the symbiotic period within 20 to 25 days, and arranging the sowing time reasonably ...	Corn intercropping technology in southern Xinjiang. Sowing date determination: It is required that the symbiotic period does not exceed corn jointing, and the symbiotic period in production is \leq 20–25 days (usually 15–20 days). Sowing is generally done on May 20–30 ...
chunk_2	What are the different types of corn varieties?	The variety types of corn can be divided into hard grain type, purslane type, powdery type, sweet type, burst type, waxy type, lemma type, and semi-purslane type according to grain morphology ...	Variety type: Corn varieties are usually classified according to grain morphology. For the convenience of understanding, research, and use, corn can also be classified according to its plant type, growth period, plant height, seed production method, grain color, and different uses ...
chunk_3	Do the three measures of post broadcast suppression, straw covering, and water retaining agents have a synergistic effect on soil moisture?	The integration of post broadcast suppression, straw covering, and water retaining agents has a significant effect on improving soil moisture ...	Figures 6-2 and 6-3 show the moisture content of different soil layers under different treatments. The trends of the two experimental points are the same. In the seedling stage, corn plants are small, and the soil moisture content of the five treatments shows D > B > C > A > E
chunk_4	What are the specific timing and types of fertilizers used for sweet corn in field management?	Fertilization of sweet corn is divided into two stages. The first time was about 10 days after transplanting, applying 15 kg of urea and 10 kg of potassium chloride every 300 square meters (0.45 acres) and conducting small soil cultivation. Second fertilization ...	Fertilizer management: If the seedling age (visible leaves) reaches 5–6 leaves during transplantation, apply 15 kg of urea and 10 kg of potassium chloride per 300 square meters (0.45 acres) of sweet corn box for about 10 days after transplantation, and cultivate the soil. About 15 days after this fertilization ...

The construction process for CornData is shown in Figure 3. Firstly, OCR text recognition technology [30,31] was used to collect professional books such as “Theory and Technology of High Yield Cultivation of Corn in Xinjiang” published by China Agricultural Science and Technology Press and “Comprehensive Manual of High Quality and High Yield Cultivation of Corn” published by Chemical Industry Press.

Python crawler technology [32–34] was used to collect data from platforms such as the Seed Industry Business Network and the China Seed Association Network as core data sources; secondly, the collected data needed to be cleaned (removing duplicate, redundant,

and invalid information), deduplicated, and standardized; finally, the logical consistency and domain relevance of the data were ensured.

CornData contains multi-level types of knowledge in the field of corn planting, such as corn variety characteristics, key cultivation techniques, pest control, etc. Compared with Internet information such as web encyclopedias, web entries, web blogs, etc., the data in this dataset is more professional in the field and is suitable for evaluating improved retrieval-augmented generation question-and-answer algorithms such as Sem-RAG. At the same time, the construction process of CornData is also very consistent with the knowledge base construction process in the field of professional knowledge Q&A, providing a feasible approach for building knowledge bases in vertical domains.

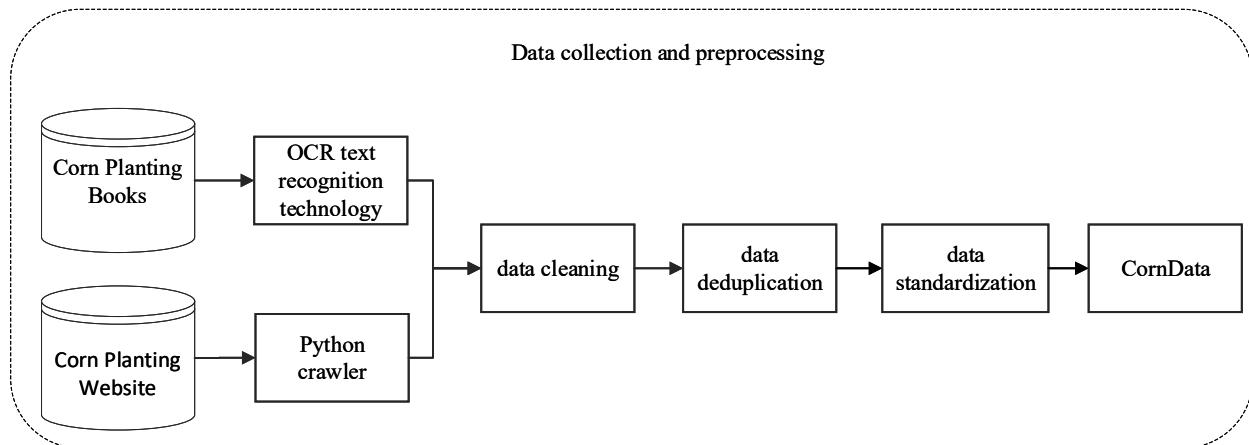


Figure 3. Data collection and preprocessing process for corn planting technology.

3.2. Experimental Environment

The operating system of the experimental environment in this article is the Linux operating system. The experimental hardware conditions are an i9-14900HX CPU and a NVIDIA GeForce RTX 4090 GPU. This article used the PyTorch (version 2.1.0) deep learning framework to conduct the experiments. The version of the PyTorch framework is 2.1.0, and the version of CUDA is 12.1.

In this study, the hyperparameters used in the LoRA fine-tuning training learning method are shown in Table 2, including learning rate, decay strategy, fine-tuning type (finetuning_type), weight decay, total epochs, and batch size.

Table 2. Initialization parameters of our method.

Parameters	Value	Description
Learning rate	1×10^{-5}	Initial learning rate
Decay strategy	Cosine	Description of the learning rate decline strategy
finetuning_type	Lora	Finetuning type
Weight decay	5×10^{-4}	The parameter settings for overfitting
Total epochs	200	The number of training rounds
Batch size	4	The capacity of every batch

In this study, the initialization parameter configuration of the proposed method is as follows: the initial value of the learning rate is set to 1×10^{-5} , and the decay strategy adopts the cosine strategy; LoRA is selected for the fine-tuning type; to alleviate overfitting, the weight decay parameter is set to 5×10^{-4} ; and during the training process, the total epochs are 200, and the batch size is set to 4.

To make our efficiency and scalability claims auditable, we evaluated each stage of the pipeline, including chunking, triple extraction, Leiden partitioning, summary generation, dual encoders, vector stores, reranking, and LoRA. For every stage, we report the per-query GFLOPs. All measurements were performed on a single NVIDIA RTX 4090 GPU with PyTorch 2.3 and CUDA 12.1, after discarding 10 warm-up runs and averaging over 10 queries. Table 3 summarizes the results.

Table 3. Initialization parameters of our method.

Stage	Params	GFLOPS
Chunking	Window = 100, chunksize = 20	0
Triple Extraction	Qwen3-8B	7731
Leiden Partition	igraph	/
Summary Generation	Qwen3-8B	7731
Dual Encoders	BGE-large	168
Vector Store	FAISS	/
Reranking	Cross-Encoder	65
LORA	Qwen3-8B	77

3.3. Experimental Metrics

In order to comprehensively evaluate the ability of the model at the level of graph nodes, this study adopted the most classic validation indicators, such as precision, recall, and F1 [35,36], with F1 as the main evaluation indicator. The calculation method for the above indicators is shown in Equations (10)–(12).

$$Precision = \frac{TP}{(TP + FP)} \quad (10)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (11)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (12)$$

Among them, TP (True Positive) represents the number of samples that the model predicts as positive and are actually positive, FP (False Positive) represents the number of samples that the model predicts as positive but are actually negative, and FN (False Negative) represents the number of samples that the model predicts as negative but are actually positive. The recall function is defined with n representing the number of samples in a certain category, and the classification accuracy of the i-th category is also defined accordingly.

To comprehensively evaluate the response capability of the algorithm, this paper adopts classical evaluation indicators such as answer correctness (Answer-C), answer relevance (Answer-R), and retrieval precision (CR) [37] in the corresponding evaluation task of question answering and uses Answer-C as the main evaluation indicator. As shown in Equations (13)–(15), the definitions of these indicators are

$$Answer - C = \frac{TP}{(TP + FP)} \quad (13)$$

$$Answer - R = \frac{A_{11} \cdot A_{21}}{\|A_{11}\| \cdot \|A_{21}\|} \quad (14)$$

$$CR = \frac{T_{taken}}{N_{total}} \quad (15)$$

Among them, Equation (13) aims to evaluate the proportion of correct answers among all the answers generated by the model. It directly reflects the accuracy and reliability of the model in specific knowledge question-answering tasks. TP (True Positive) refers to the number of samples in which the model provides correct and factual responses. FP (False Positive) refers to the number of samples with incorrect or untrue responses provided by the model.

Equation (14) is a variant of cosine similarity used to calculate the similarity between vectors. A_{11} and A_{21} represent row and column vectors in two vectors. $A_{11}A_{21}$ represents the dot product of vectors, measuring their consistency in direction. $\|A_{11}\| \|A_{21}\|$ represents the norm of a vector, which is its length.

Equation (15) is used to calculate the text citation ratio. This indicator aims to measure the proportion of cited parts in an article. The numerator T_{take} represents the actual number of characters used in this article, and the denominator N_{total} represents the number of reference characters provided in this article.

3.4. Experimental Results

3.4.1. Evaluation of Graph Node Extraction

Graph nodes refer to the entities in each text chunk. Due to the stylistic deviation between the entities and entity types extracted from entity extraction and the dataset itself, in order to objectively evaluate the ability of different language base models in extracting graph nodes, this paper uses LoRA's algorithm to fine-tune a small number of samples extracted from different language base models on the training and validation sets of CLUENER2020 so that different language base models can extract entity information consistent with the style of the dataset itself in entity extraction tasks. The fine-tuned large language model was subjected to entity extraction testing on the CLUENER2020 test set, and the performance of different large language models was compared with that of traditional small models used for entity extraction. The test results are shown in Tables 4 and 5.

Table 4. Extraction performance of different methods on the CLUENER2020 dataset.

Models	P%	R%	F1%
Human Performance	65.74	62.17	63.41
BILSTM-CRF	71.06	68.97	70.00
BERT	77.24	80.46	78.82
RoBERTa	79.26	81.69	80.42
Baichuan2-7B	70.88	68.48	69.36
Chatglm3-6B	72.24	66.94	69.28
Llama3-8B	74.24	71.38	72.58
Qwen3-8B (ours)	75.45	73.45	74.44

As shown in Table 4, different models perform differently in extracting entities from the CLUENER2020 dataset. Among them, the performance of humans manually extracting entities is the worst, while traditional small models such as BILSTM-CRF [38], BERT [39], and RoBERTa [40] perform well in the accuracy P, recall R, and F1 metrics. RoBERTa is the model with the best performance in entity extraction tasks among all models. However, small models represented by RoBERTa have limited parameters and require extensive training based on datasets from different fields, which cannot meet the needs of general entity extraction. Pre-trained large language models perform well in general tasks due to their large number of parameters. Therefore, this article selected pre-trained open-source large language models represented by Baichuan2-7B, Chatglm3-6B, Llama3-8B, and Qwen3-8B for the task of graph node extraction. According to Table 5, compared to other large

language models, Qwen3-8B performs the best in accuracy, recall, and F1. Meanwhile, compared to traditional small models, Qwen3-8B is similar to RoBERTa in terms of the accuracy, recall, and F1 metrics. This indicates that Qwen3-8B has a good entity extraction capability on the Chinese dataset CLUENER2020, while also meeting the requirements for the general extraction of graph nodes. Therefore, the base language model of the Sem-RAG model proposed in the study is chosen as Qwen3-8B.

Table 5. Performance of different models in extracting entities of different categories on the CLUENER2020 dataset.

Entity Type	BILSTM-CRF F1%	BERT F1%	RoBERTa F1%	Baichuan2-7B F1%	Chatglm3-6B F1%	Llama3-8B F1%	Qwen3-8B (Ours) F1%
Person Name	74.04	88.75	89.09	83.31	84.51	87.60	87.03
Organization	75.96	79.43	82.34	75.47	72.31	74.16	76.44
Position	70.16	78.89	79.62	69.90	66.18	71.96	78.11
Company	72.27	81.42	83.02	71.03	71.48	73.25	74.20
Address	45.50	60.89	62.63	52.61	51.64	54.09	56.30
Game	85.27	86.42	86.80	81.72	79.65	82.28	83.55
Government	77.25	87.03	88.17	72.78	77.25	77.75	78.20
Scene	52.42	65.10	70.49	47.79	48.59	56.54	61.93
Book	67.20	73.68	74.60	66.67	73.05	73.49	72.26
Movie	78.97	85.82	87.46	72.34	68.13	74.73	76.40

As shown in Figure 4, during fine-tuning training, the loss of Qwen3-8B decreases steadily as the training phase progresses. On the other hand, the loss of Qwen3-8B ultimately decreased to a level close to 0, indicating that fine-tuning training had a better effect. This indicates that the model's preference for an entity output format gradually aligns with the style provided by the labels in the dataset, and the alignment effect is good.

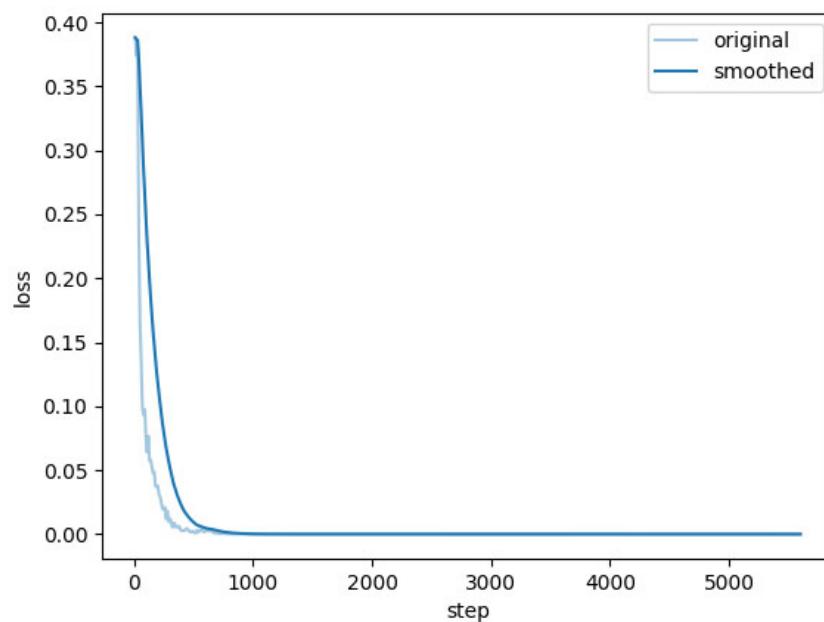


Figure 4. Entity extraction fine-tuning experiment for graph node extraction evaluation.

To assess the reliability of our triple extraction process, we conducted an evaluation on a hand-labeled subset of the data. We randomly sampled sentences and annotated their gold-standard triples. We then compared these annotations with the triples extracted by our model (Qwen3-8B).

Table 6 presents an illustrative example based on a corn-related paragraph. The table contrasts the gold triples with the model outputs, showing both correct and partially

incorrect extractions. As shown in the table, Qwen3-8B extracted 10 triples, of which 7 matched the gold standard, while 3 contained errors (incomplete information or missing modifiers). Two gold triples were not captured by the model. This yields precision = 70.0%, recall = 77.8%, and F1 = 73.7%.

To further clarify how extraction errors propagate to downstream summaries, we constructed an error-path table (Table 7). The analysis reveals two main error types: (i) incomplete triples that underestimate key information and (ii) modifier loss or spurious triples that distort meaning. These findings show that while triple extraction is generally reliable, residual errors may lead to subtle distortions in community-level summaries.

Table 6. The results of triple extraction.

Original Text	Gold Triples	Qwen3-8B Triples
Corn is one of the most important food crops in the world and originated in Mexico.	(Corn, originated in, Mexico), (Corn, is, one of the most important food crops in the world)	(Corn, originated in, Mexico), (Corn, is, a food crop)
It is not only an important source of human food but also widely used for feed and industrial production.	(Corn, used for, human food), (Corn, used for, feed), (Corn, used for, industrial production)	(Corn, used for, human food), (Corn, used for, feed), (Corn, used for, industrial production)
Studies show that corn yield declines significantly under heat and drought conditions.	(Heat and drought conditions, affect, corn yield decline)	(Heat and drought, cause, corn yield decline), (Corn, declines under, climate conditions)
In recent years, Northeast China has improved corn yields by adopting improved varieties and promoting water-saving irrigation.	(Northeast China, improved, corn yield) (Northeast China, method, improved varieties) (Northeast China, method, water-saving irrigation)	(Northeast China, improved, corn yield), (Northeast China, used, improved varieties) (Northeast China, used, irrigation)

Table 7. Error path table: extraction errors and their impact on summaries.

Error Type	Extracted Triple	Correct Triple	Impact on Summary
Incomplete information	(Corn, is, a food crop)	(Corn, is, one of the most important food crops in the world)	The summary may downplay the central role of corn in global food security.
Modifier loss	(Northeast China, used, irrigation)	(Northeast China, method, water-saving irrigation)	The summary may incorrectly imply that only standard irrigation was used, omitting the contribution of water-saving technology.
Spurious relation	(Corn, declines under, climate conditions)	(Heat and drought conditions, affect, corn yield decline)	The summary may overgeneralize by attributing yield decline to “climate” broadly instead of specific conditions.

3.4.2. Q&A Response Evaluation

The effectiveness of Q&A responses reflects the ability of dialogue models to provide decision-making knowledge in vertical domains. In order to evaluate the question-and-answer response ability of the proposed Sem-RAG algorithm, the classic retrieval-enhanced generation algorithm was selected as the comparative experimental group. The comparative experimental group and the Sem-RAG algorithm group were validated and evaluated on CornData. The algorithms used in the comparative experimental group include Self-RAG, Speculative-RAG, NaiveRAG, GraphRAG, the experimental group evaluated solely through LoRA fine-tuning, and the experimental group combining Sem-RAG with LoRA fine-tuning. In the experiment of Q&A response evaluation, the evaluation questions were focused on issues related to corn planting technology. The experimental results of the

evaluation are shown in Table 8. In addition to the CR retrieval evaluation metric, we computed a source-support metric, defined as the proportion of factual spans directly supported by retrieved chunks or summaries. Furthermore, we conducted a human audit on 80 randomly sampled Q&A pairs. Two annotators independently judged the factual support. As shown in the revised Table 8, our methods consistently achieve higher source-support scores than other models, confirming the reliability of the CR measure while providing stronger evidence of retrieval faithfulness.

Table 8. Q&A response metrics of mainstream LLM on CornData.

LLM	Answer-C	Answer-R	CR	Source-Support
Qwen3-8B	40.00	82.00	/	/
Qwen3-8B+LoRA	80.85	75.50	/	/
Qwen3-8B+Self-RAG	93.50	81.80	69.20	61.0
Qwen3-8B+Speculative-RAG	93.60	82.60	68.60	61.5
Qwen3-8B+NaiveRAG	92.00	82.90	68.80	60.8
Qwen3-8B+GraphRAG	92.90	83.80	69.60	62.7
Qwen3-8B+Sem-RAG	94.60	84.60	70.40	64.0
Qwen3-8B+LoRA+Sem-RAG	95.10	85.25	71.45	65.2

As shown in Table 8 and Figure 5, all experimental groups used Qwen3-8B as the base model, and the effectiveness of the question-and-answer responses varied among different experimental groups due to different strategies for enhancing retrieval. Among them, the experimental group that only responded through the baseline model had only 40% of the Answer-C index, indicating that although the baseline model has a certain general response ability, it lacks knowledge in the field of corn planting and cannot accurately respond to vertical domain problems. The Sem-RAG group performed the best in the algorithm group enhanced solely by external knowledge, with Answer-C, Answer-R, and CR reaching 94.6%, 84.6%, and 70.4%, respectively. This stems from the fact that Sem-RAG enhances its generation ability by integrating surface semantics and contextual semantic information from the field of corn cultivation, enabling Sem-RAG to accurately extract reference information sources from external knowledge and better respond to user questions. On the basis of the Sem-RAG algorithm, the performance of the groups was slightly enhanced by combining LoRA fine-tuning, with Answer-C, Answer-R, and CR improving by 0.5%, 0.65%, and 0.85%.

The Sem-RAG experimental group demonstrated an outstanding overall algorithmic performance, while the performance of the other experimental groups was inferior to that of the Sem-RAG group. As shown in Table 8 and Figure 6, although Self-RAG and Speculative-RAG achieved Answer-C scores of 93.5% and 93.6%, respectively, indicating considerable accuracy in question responses, their retrieval accuracy (CR) was relatively poor, suggesting that the retrieved reference source information was relatively lengthy. In addition, although NaiveRAG and GraphRAG performed evenly across the Answer-C, Answer-R, and CR metrics, NaiveRAG only utilized the surface semantics in the knowledge base for similar chunk matching and lacked processing of the contextual semantic information, leading to a poor performance in scenarios with complex contextual semantics. GraphRAG, while considering the processing of contextual semantic information, excessively captured the triple association semantics in the knowledge base, resulting in a poor performance in data scenarios where semantic information was scarce and failing to coordinate the relationship between triple association semantics and surface semantics. Therefore, in the comparative experiments, Sem-RAG performed the best. This also validates the rationality of integrating contextual association semantics with surface semantics to enhance the model's response ability in the field of corn planting knowledge.

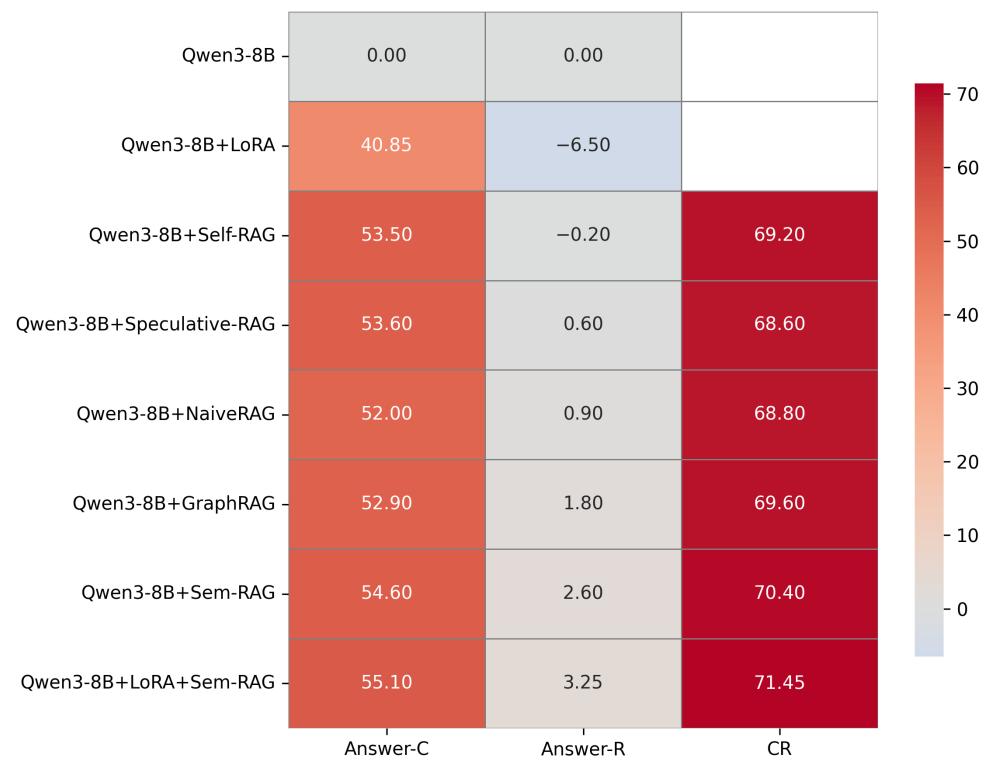


Figure 5. A heatmap of the performance improvements across different experimental groups relative to the baseline model group for each metric.

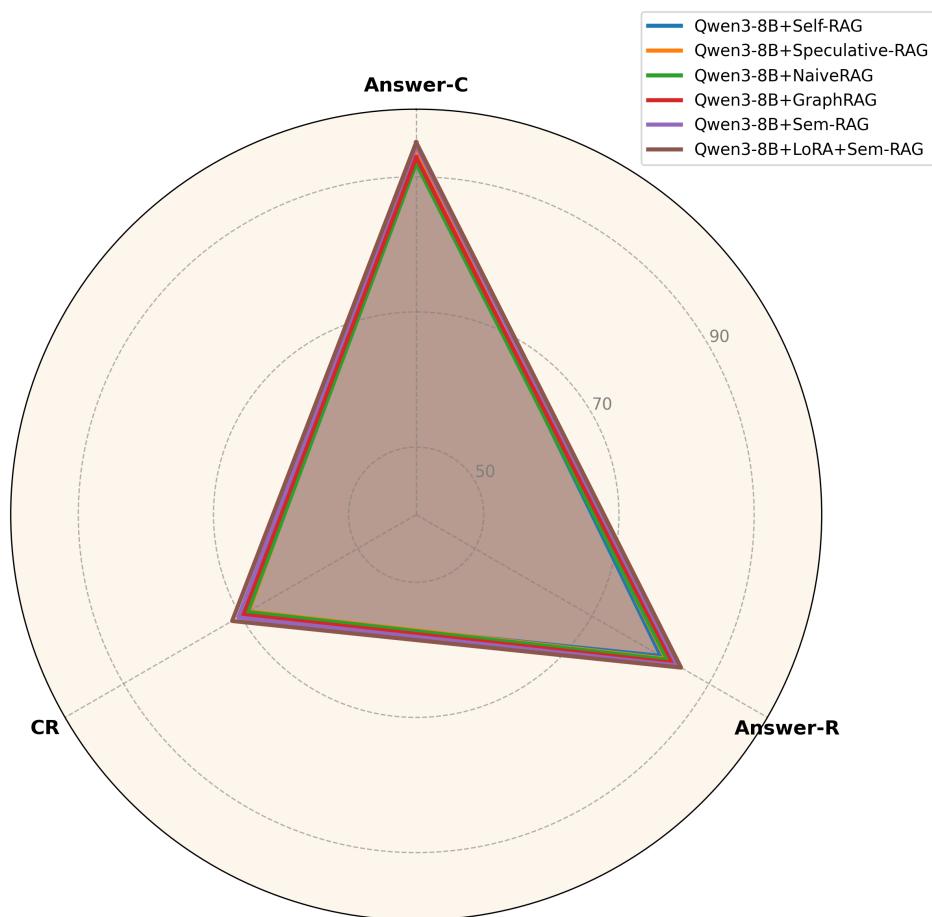


Figure 6. Overall performance of different experimental groups across all metrics.

To further explore how the number and granularity of communities affected question-answering accuracy, we adjusted the parameter nodes, which limited the number of community nodes, and examined the response evaluation results of the Graph-Human model on the CornData dataset. The specific experimental results are shown in Table 9.

Table 9. The relationship between community number and granularity and question-answering accuracy.

Id	Nodes	Community	Answer-R%	Answer-C%	Answer-S%
1	5	[4, 20]	88.30	75.20	65.10
2	10	[2, 10]	94.60	84.60	70.40
3	20	[1, 5]	92.10	82.30	68.50
4	50	[1, 2]	89.40	78.10	66.20

To examine the impact of the Leiden community detection parameters, we varied the number of nodes and the community size range. As shown in Table 9, the performance is relatively stable across different settings, with the best results obtained when using 10 nodes and a community size range of [2, 10]. This suggests that our method is robust to the choice of the Leiden parameters, while the default configuration provides a good balance between recall (Answer-R), correctness (Answer-C), and overall score (Answer-S).

As shown in Table 10, the comparison of the Q&A examples between Sem-RAG and other models can be observed. Analysis of the data in Table 10 indicates that using Qwen3-8B alone as the model for the Q&A task allows the model to answer the description of the corn jointing stage; however, the description primarily reflects empirical observations in practical production and is not comprehensive. The experimental group combining Qwen3-8B with NaiveRAG exhibits a similar issue, merely presenting information about the jointing stage retrieved from surface-level semantics in the reference texts. Likewise, the experimental groups combining Self-RAG and Speculative-RAG also show this limitation, focusing mainly on empirical descriptions of the jointing stage in agricultural production while adding some insights on its significance.

In contrast, the experimental group combining Sem-RAG and GraphRAG provides a relatively comprehensive answer regarding the corn jointing stage, fully explaining its definition, duration, and significance and practical experience in agricultural production. This demonstrates that deeper semantic information plays a significant role in retrieval-augmented generation responses. Compared with the GraphRAG experimental group, Sem-RAG's responses are more tightly focused on the definition, timing, developmental phase, key significance, and agricultural relevance of the corn jointing stage. This indicates that Sem-RAG effectively exploits the semantic information in the reference texts to generate responses. Therefore, the Sem-RAG experimental group shows a clear advantage in experimental performance.

All responses presented were reviewed and validated by two agricultural experts with more than ten years of experience in crop science. The experts assessed the accuracy and completeness of the model-generated answers, ensuring that the comparative analysis reflected domain-verified responses.

Beyond model comparison (Table 10), we further categorized the errors observed in Sem-RAG into three types: incorrect agronomic conditions, wrong timing windows, and mislocalized regions. Table 11 presents representative Q&A examples for each category, together with the correct answers and the error sources (chunk matching vs. community summarization).

Table 10. Comparison of Q&A examples between Sem-RAG and other models.

Model	Question	Response
Qwen3-8B	How is the jointing stage of corn defined?	The jointing stage of corn is defined as the date when more than 60% of the plants in the field have a main stem height of approximately 3 cm. At this stage, the basal part of the plant becomes rounded, stem nodes can be palpated, the fifth internode of the main stem begins to elongate, and the tassel growth cone also starts to develop.
Qwen3-8B+ NaiveRAG	How is the jointing stage of corn defined?	The jointing stage of corn is commonly defined as the date when more than 60% of the plants in the field have a main stem height of approximately 3 cm.
Qwen3-8B+ Self-RAG	How is the jointing stage of corn defined?	In the growth process of corn, the jointing stage refers to the period when the internodes at the base of the stem begin to elongate rapidly. It is usually identified as the date when more than 60% of the plants in the field have visibly elongated main stem internodes of about 3 cm. This stage is critical because the plant enters the transition from vegetative growth to reproductive growth; the stem elongates rapidly, leaf area expands quickly, and the demand for water and nutrients—particularly nitrogen—increases significantly, thereby marking the onset of the key period for yield formation.
Qwen3-8B+ Speculative-RAG	How is the jointing stage of corn defined?	In the entire growth cycle of corn, the jointing stage marks the period when the basal internodes of the stem enter a phase of rapid elongation. It is generally identified as the time when more than 60% of the plants in the field have visibly elongated main stem internodes of approximately 3 cm. This stage is considered critical because the plant is gradually transitioning from vegetative to reproductive growth; the stem elongates more rapidly, leaf area expands sharply, the demand for water and nutrients—especially nitrogen—rises significantly, and it also signals the formal onset of the key period for yield formation.
Qwen3-8B+ GraphRAG	How is the jointing stage of corn defined?	The jointing stage of corn specifically refers to the growth phase during which the plant begins to elongate rapidly from the basal stem and develops distinct internodes. This stage occurs after seedling emergence and belongs to the vegetative growth phase. It marks the transition of corn from the seedling stage to a period of rapid growth and represents a critical point in its developmental process. The jointing stage typically occurs 10 to 15 days before tasseling and plays a vital role in determining the final yield of corn. In practice, it is usually identified as the date when more than 60% of the plants in the field have a main stem height of approximately 3 cm.
Qwen3-8B+ Sem-RAG	How is the jointing stage of corn defined?	The jointing stage of corn refers to a critical growth phase in which the basal stem begins to elongate significantly and visible internodes are formed. This stage occurs after seedling emergence and still belongs to the vegetative growth phase. The onset of the jointing stage indicates the transition of corn from the seedling phase to a period of rapid growth, serving as an important turning point in its overall development. It typically occurs 10 to 15 days before tasseling and plays a decisive role in determining the final yield. In practical production, the jointing stage is commonly identified as the time when more than 60% of the plants in the field have a main stem height of approximately 3 cm.
Qwen3-8B+ LoRA + Sem-RAG	How is the jointing stage of corn defined?	The jointing stage of corn refers to a critical developmental phase during which the basal stem begins to elongate markedly and visible internodes are formed. This stage occurs after seedling emergence and still belongs to the vegetative growth phase. The onset of the jointing stage signifies the transition of corn from the seedling phase to a period of rapid growth, representing an important turning point in its overall development. It usually occurs 10 to 15 days before tasseling and plays a decisive role in determining the final yield. In practical production, the jointing stage is commonly identified as the time when more than 60% of the plants in the field have a main stem height of approximately 3 cm.

Table 11. Failure taxonomy of Q&A errors in *Sem-RAG*.

Failure Type	Question	Model Response (Sem-RAG)	Correct Answer	Error Source
Incorrect agronomic condition	What crop grows best in Region A?	“Rice is the main crop in Region A.”	Maize is the major crop due to local soil and climate conditions.	Chunk match (retrieval selected docs on rice instead of maize)
Wrong timing window	When should wheat be planted in Region B?	“April is the recommended sowing period.”	June–July is the correct sowing window.	Community summary (misinterpreted retrieved guidance)
Mis-localized region	What are the irrigation practices in Northeast China?	“Flood irrigation is widely used in North China Plain.”	Water-saving irrigation (sprinkler/drip) is prevalent in Northeast China.	Chunk match (retrieval pulled passages from the wrong region)

3.4.3. Ablation Experiment

The study used the corn cultivation knowledge domain dataset CornData to explore the performance of the NaiveRAG, GraphRAG, and Sem-RAG algorithms within a base large language model. The experimental results are shown in Table 12.

Table 12. The ablation experiments on CornData.

Methods	Answer-C%	Answer-R%	CR%
LLM	40.00	82.00	/
LLM+LoRA	80.85	75.50	/
NaiveRAG	92.00	82.90	68.80
LLM+LoRA+NaiveRAG	93.60	83.60	69.80
GraphRAG	92.90	83.80	69.60
LLM+LoRA+GraphRAG	94.50	84.20	70.20
Sem-RAG	94.60	84.60	70.40
LLM+LoRA+Sem-RAG	95.10	85.25	71.45

The base LLM performed poorly in the Answer-C metric, achieving only 40%, indicating that an unoptimized model struggles to meet the requirements of corn domain tasks. Sem-RAG outperformed both GraphRAG and NaiveRAG across all metrics. For the Answer-C metric, Sem-RAG (94.60%) improved by 1.70 percentage points over GraphRAG (92.90%) and by 2.60 percentage points over NaiveRAG (92.00%); for the Answer-R metric, Sem-RAG (84.60%) was 0.80 percentage points higher than GraphRAG (83.80%) and 1.70 percentage points higher than NaiveRAG (82.90%); for the CR metric, Sem-RAG (70.40%) exceeded GraphRAG (69.60%) by 0.80 percentage points and NaiveRAG (68.80%) by 1.60 percentage points.

Compared with the base LLM, applying LoRA tuning alone improves Answer-C from 40.00% to 80.85%, showing a large gain in factual accuracy, although Answer-R slightly decreases. This confirms that LoRA is the main factor in adapting the LLM to the corn domain. When further combined with retrieval methods (NaiveRAG, GraphRAG, Sem-RAG), the gains are complementary: retrieval design improves robustness and coverage, while LoRA provides domain adaptation. Due to computational constraints, we did not extensively vary the LoRA rank r . We fixed $r = 8$ based on preliminary tests and prior work, where the performance typically saturates when $r \geq 8$.

This series of results fully demonstrates that in corn domain tasks, the Sem-RAG strategy with semantic enhancement can utilize external knowledge more effectively and improve the task performance compared with that of the graph-structure-based GraphRAG and the simple retrieval-based NaiveRAG strategies.

The ablation experiment also revealed the performance differences between different experimental groups in the Answer-C and Answer-R dimensions. As shown in Figure 7, the Answer-C level of the baseline LLM is significantly lower, and although fine-tuning alone can improve Answer-C, the decrease in Answer-R reflects the insufficient generalization ability. In contrast, NaiveRAG was able to achieve a relatively balanced improvement in both indicators, indicating that the retrieval enhancement mechanism has a significant increase in model performance. On this basis, combining fine-tuning with RAG technology, such as LoRA+NaiveRAG and LoRA+GraphRAG, further improved the stability and overall performance of the model. Especially with the introduction of the Sem-RAG experimental group based on semantic enhancement and the LoRA+Sem-RAG experimental group, the best results were achieved in both the Answer-C and Answer-R dimensions, highlighting the significant advantages of semantic retrieval mechanisms in improving the quality and consistency of responses in large language models.

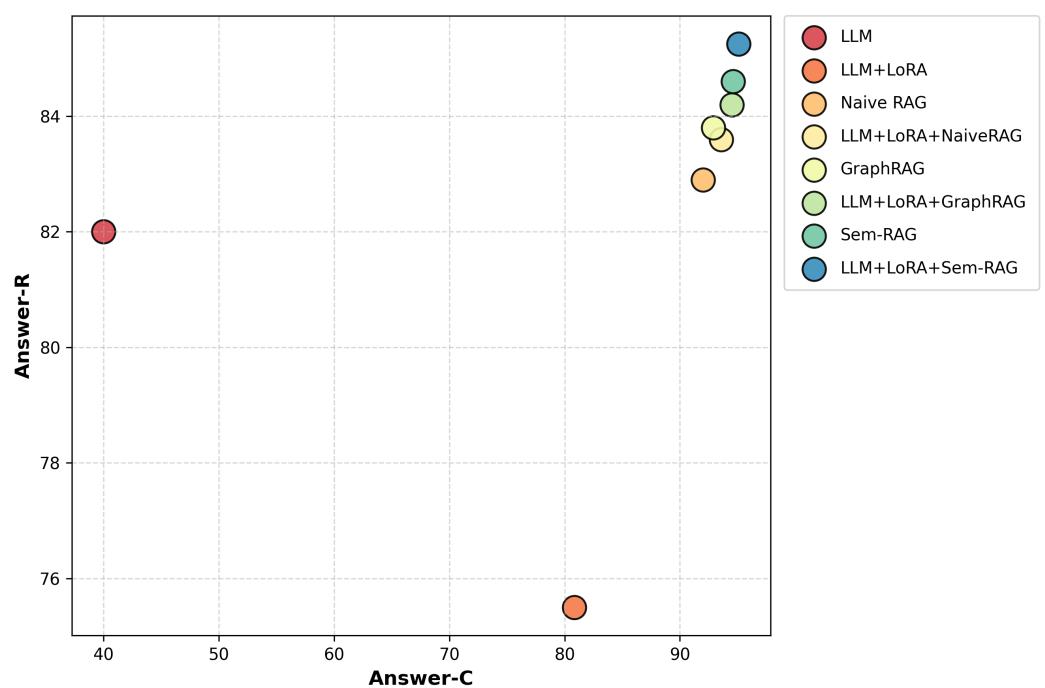


Figure 7. The relationship between Answer-R and Answer-C across different experimental groups.

We also conducted an ablation study on the community summarization component to evaluate the contribution of different fields. Specifically, we compared three settings, (i) topic-only, (ii) topic + key relations, and (iii) full summarization with topic, key relations, conditions, and steps, as detailed in Table 13.

As shown in Table 14, using only topic information leads to relatively low Answer-C scores, while adding key relations already brings a substantial improvement. Incorporating conditions and steps further enhances the performance, reaching the best results consistent with the full Sem-RAG model. This indicates that key relations are the most influential field, and conditions and steps provide complementary benefits for improving the precision and completeness of answers.

Table 13. Distribution of different fields in community summarization.

ID	Topic	Key Relations	Conditions	Steps
1	Corn Planting Time	Planting → Temperature Requirements	Local soil temperature stabilizes above 10 °C	Seed selection → Land preparation → Timely planting
2	Corn Fertilization Requirements	Growth Period → Nitrogen/Phosphorus/Potassium Demand	Under different soil fertility levels, nitrogen demand varies significantly	Base fertilizer application at seedling stage → Nitrogen topdressing at jointing stage → Potassium supplementation at the bell stage
3	Corn Pest and Disease Control	Pests and Diseases → Main Control Methods	In high-temperature and high-humidity environments in the south, leaf blight is more common	Select disease-resistant varieties → Seed coating → Spraying pesticides during the growth period
4	Corn Irrigation Management	Irrigation → Key Periods	Especially critical in arid and low-rainfall regions	Light watering at seedling stage → Focused irrigation at tasseling and silking stage → Moderate watering at grain filling stage
5	Corn Harvest Timing	Harvest → Moisture Content and Yield	When grain moisture content drops to around 25%	Field observation of milk line disappearance → Mechanical harvesting → Timely drying

Table 14. Effects of different community summarization fields on Answer-C.

LLM	Community Summary Setting	Answer-C (%)
Qwen3-8B+Sem-RAG	Topic-only	87.20
Qwen3-8B+Sem-RAG	Topic+Key relations	91.50
Qwen3-8B+Sem-RAG	Full(Topic+Rel+Cond+Steps)	94.60

4. Conclusions

This study aims to address the issues of hallucination and insufficient accuracy in retrieval-generated responses of LLMs and RAG techniques in knowledge-intensive scenarios, particularly in agriculture. To enhance the accuracy and reliability of corn planting knowledge question-answering tasks, a fine-grained semantic information retrieval-enhanced algorithm, named Sem-RAG, is proposed. This algorithm first divides professional documents on corn planting knowledge into fixed-length chunks; then, it extracts semantically associated triples from each chunk; next, the semantic triples in each chunk are organized into communities using the Leiden algorithm, and thematic community summaries are generated; afterwards, the original text of each chunk and the thematic summaries are separately vectorized to construct a surface semantic vector knowledge base and a fine-grained semantic vector knowledge base for retrieval; then, the user query vector is first used to perform similar-chunk retrieval on the surface semantic knowledge base, and through these similar chunks, the corresponding graph community summaries in the fine-grained semantic knowledge base are located; finally, the information from similar chunks and the graph community summaries is used as semantic reference for the LLM to generate the query response.

The algorithm was evaluated on CornData for corn knowledge question-answering. Its Answer-C, Answer-R, and CR scores reached 94.6%, 84.6%, and 70.4%, respectively, which were 2.6%, 1.7%, and 1.6% higher than those of the traditional NaiveRAG method. The results indicate that Sem-RAG effectively combines surface semantic information with

contextual semantic association information, significantly improving the performance in agricultural knowledge question-answering tasks.

While Sem-RAG demonstrates strong retrieval and reasoning capabilities across diverse benchmarks, several limitations remain. First, the model's performance may degrade when the input data are noisy, ambiguous, or incomplete, as the semantic retrieval module relies on high-quality representations. Future work could explore noise-robust embedding strategies or adaptive confidence mechanisms to mitigate this issue. Second, although our method scales reasonably well with medium-sized corpora, the computational costs increase with corpus size due to the semantic similarity computations required during retrieval. Techniques such as approximate nearest neighbor (ANN) search, vector compression, or hierarchical retrieval could improve scalability and efficiency. Finally, our current evaluation primarily focuses on static datasets; extending Sem-RAG to dynamic or streaming contexts remains an open challenge.

In the future, we plan to further investigate potential hardware deployment issues that may arise in practical applications. In addition, improving the algorithmic performance of corn planting knowledge question-answering through multi-agent collaboration is an important aspect to consider. We also aim to explore extending purely text-based knowledge question-answering to image and video modalities to enhance the model performance, leveraging technologies such as diffusion models and Sora.

Author Contributions: Conceptualization: B.B. and X.M.; methodology: B.B.; software: B.B.; validation: B.B., X.M.; formal analysis: B.B.; investigation: B.B.; resources: B.B. and C.Z.; data curation: B.B. and C.Z.; writing—original draft preparation: B.B.; writing—review and editing: B.B. and X.M.; visualization: C.Z.; supervision: X.M.; project administration: X.M.; funding acquisition: X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key R&D Program of China (2022ZD0115805) and the Provincial Key S&T Program of Xinjiang (2022A02011).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2017; pp. 5998–6008.
2. Rogers, A.; Kovaleva, O.; Rumshisky, A. A primer in BERTology: What we know about how BERT works. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 842–866. [[CrossRef](#)]
3. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
4. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 5485–5551.
5. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.* **2023**, *24*, 11324–11436.
6. Yang, H.; Chen, H.; Guo, H.; Li, X.; Wang, Y.; Zhang, Y.; Liu, Y.; Xie, X. Llm-medqa: Enhancing medical question answering through case studies in large language models. *arXiv* **2024**, arXiv:2501.05464.
7. Salim, M.S.; Hossain, S.I.; Jalal, T.; Hasan, M.A.; Shin, J. LLM based QA chatbot builder: A generative AI-based chatbot builder for question answering. *SoftwareX* **2025**, *29*, 102029. [[CrossRef](#)]

8. Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* **2025**, *43*, 1–55. [[CrossRef](#)]
9. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2020; pp. 9459–9474.
10. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-augmented generation for large language models: A survey. *arXiv* **2023**, arXiv:2312.10997. [[CrossRef](#)]
11. Fang, Y.; Zhan, J.; Ai, Q.; Chen, J.; Ouyang, S.; Zhao, W.X. Scaling laws for dense retrieval. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, DC, USA, 14–18 July 2024; pp. 1339–1349.
12. Fan, W.; Ding, Y.; Ning, L.; Wang, S.; He, Y.; Wang, Y.; Yin, D.; Wang, C.; Zhang, D. A survey on rag meeting llms: Towards retrieval-augmented large language models. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024; pp. 6491–6501.
13. Du, Y.; Jin, X.; Yan, R.; Zhao, J.; Zhang, X.; Wang, Y. Sentiment enhanced answer generation and information fusing for product-related question answering. *Inf. Sci.* **2023**, *627*, 205–219. [[CrossRef](#)]
14. Tam, D.; Menon, R.R.; Bansal, M.; Srivastava, S.; Raffel, C. Improving and simplifying pattern exploiting training. *arXiv* **2021**, arXiv:2103.11955. [[CrossRef](#)]
15. Oche, A.J.; Folashade, A.G.; Ghosal, T.; Saha, S.; Ekpo, R.H. A systematic review of key retrieval-augmented generation (rag) systems: Progress, gaps, and future directions. *arXiv* **2025**, arXiv:2507.18910. [[CrossRef](#)]
16. Zakka, C.; Shad, R.; Chaurasia, A.; Park, S.J.; Kim, C.; Dalal, A.; Hiesinger, W.; Shademan, A.; Morris, M.X.; Fong, R.; et al. Almanac—Retrieval-augmented language models for clinical medicine. *NEJM AI* **2024**, *1*, A10a2300068. . [[CrossRef](#)]
17. Malali, N. The Role of Retrieval-Augmented Generation (RAG) in Financial Document Processing: Automating Compliance and Reporting. *Int. J. Manag.* **2025**, *12*, 26–46. [[CrossRef](#)]
18. Su, W.; Tang, Y.; Ai, Q.; Mao, J.; Liu, Y.; Zhao, W.X. Parametric retrieval augmented generation. In Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, Padua, Italy, 13–18 July 2025; pp. 1240–1250.
19. Ding, W.; Cao, Y.; Zhao, D.; Bärgman, J.; Huang, C. Realgen: Retrieval augmented generation for controllable traffic scenarios. In *European Conference on Computer Vision*; Springer Nature: Cham, Switzerland, 2024; pp. 93–110.
20. Wang, H.; Prasad, A.; Stengel-Eskin, E.; Bansal, M. Retrieval-augmented generation with conflicting evidence. *arXiv* **2025**, arXiv:2504.13079. [[CrossRef](#)]
21. Béchard, P.; Ayala, O.M. Reducing hallucination in structured outputs via Retrieval-Augmented Generation. *arXiv* **2024**, arXiv:2404.08189. [[CrossRef](#)]
22. Shao, Z.; Gong, Y.; Shen, Y.; Huang, M.; Duan, N.; Chen, W. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv* **2023**, arXiv:2305.15294.
23. Ovadia, O.; Brief, M.; Mishaeli, M.; Elisha, O. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv* **2023**, arXiv:2312.05934.
24. Amugongo, L.M.; Mascheroni, P.; Brooks, S.; Bidarmaghz, A.; Kifle, M.; Kaluza, B.; Celik, T. Retrieval augmented generation for large language models in healthcare: A systematic review. *PLoS Digit. Health* **2025**, *4*, e0000877.
25. Zhuang, Y.; Yu, Y.; Wang, K.; Sun, H.; Shi, C. Toolqa: A dataset for llm question answering with external tools. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation, Inc. (NeurIPS): La Jolla, CA, USA, 2023; pp. 50117–50143.
26. Wu, J.; Zhu, J.; Qi, Y.; Sun, Z.; He, X.; Xie, X.; Xu, H.; Chen, Y.; Zhang, Y.; Xing, Z.; et al. Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. *arXiv* **2024**, arXiv:2408.04187. [[CrossRef](#)]
27. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. Lora: Low-rank adaptation of large language models. In Proceedings of the 2022 International Conference on Learning Representations, Online, 25–29 April 2022.
28. Li, Y.; Yu, Y.; Liang, C.; He, P.; Karampatziakis, N.; Chen, W.; Zhao, T. Loftq: Lora-fine-tuning-aware quantization for large language models. *arXiv* **2023**, arXiv:2310.08659.
29. Mao, Y.; Ge, Y.; Fan, Y.; Ma, C.; Wang, Z.; Sun, R.; Lou, J.G.; Zhang, D. A survey on lora of large language models. *Front. Comput. Sci.* **2025**, *19*, 197605. [[CrossRef](#)]
30. Pal, U.; Halder, A.; Shivakumara, P.; Lu, T.; Blumenstein, M.; Lopresti, D. A Comprehensive Review on Text Detection and Recognition in Scene Images. *Artif. Intell. Appl.* **2024**, *2*, 229–249.
31. Sharma, P. Advancements in OCR: A deep learning algorithm for enhanced text recognition. *Int. J. Inven. Eng. Sci.* **2023**, *10*, 1–7. [[CrossRef](#)]
32. Brin, S.; Page, L. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.* **1998**, *30*, 107–117. [[CrossRef](#)]

33. Mirtaheri, S.M.; Dinçtürk, M.E.; Hooshmand, S.; Bochmann, G.V.; Jourdan, G.V.; Onut, I.V. A brief history of web crawlers. *arXiv* **2014**, arXiv:1405.0749. [[CrossRef](#)]
34. Lotfi, C.; Srinivasan, S.; Ertz, M.; Latrous, I. Web Scraping Techniques and Applications: A Literature. In *SCRS Conference Proceedings on Intelligent Systems*; SCRS Publications: New Delhi, India, 2021; pp. 381–394.
35. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *European Conference on Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
36. Saracevic, T. Evaluation of evaluation in information retrieval. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, 9–13 July 1995; pp. 138–146.
37. Es, S.; James, J.; Anke, L.E.; Schockaert, S. Ragas: Automated evaluation of retrieval augmented generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, St. Julians, Malta, 17–22 March 2024; pp. 150–158.
38. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360. [[CrossRef](#)]
39. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
40. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.