TreeRAG: Unleashing the Power of Hierarchical Storage for Enhanced Knowledge Retrieval in Long Documents

Anonymous ACL submission

Abstract

When confronting long document information retrieval for Query-Focused Summarization(QFS), Traditional Retrieval-Augmented Generation(RAG) frameworks struggle to retrieve all relevant knowledge points, and the chunking and retrieve strategies of existing frameworks may disrupt the connections between knowledge points and the integrity of the information. To address these issues, we propose **TreeRAG**, which employs **Tree-Chunking** for chunking and embedding in a tree-like structure , coupled with "**root-to-leaves**" and "**leaf-to-root**" retrieve strategy named **Bidirectional Traversal Retrieval**. This approach effectively preserves the hierarchical structure among knowledge points and significantly enhances the ability to retrieve while minimizing noise inference. Our experimental results on the **Finance, Law, and Medical subsets of the Dragonball dataset** demonstrate that **TreeRAG** achieves significant enhancements in both recall quality and precision compared to traditional and popular existing methods and achieves better performance to corresponding question-answering tasks, marking a new breakthrough in long document knowledge retrieval.

1 Introduction

In the domain of Natural Language Processing(NLP). RAG, initially proposed by Lewis et al. (2021), has emerged as a pivotal strategy for enhancing the text generation capabilities of Large Language Models(LLMs) by integrating information from external knowledge bases, leading to outstanding performance across a variety of NLP tasks (Ji et al., 2023; Izacard and Grave, 2021; Borgeaud et al., 2022). This technique incorporates specialized books or documents related to particular domain into the knowledge base, thereby enhancing domain-specific expertise and accuracy of model in specific fields.

Across various general domains, with the increase of knowledge base content due to iteration or the emergence of large-scale documents as knowledge base content, structured or semi-structured long documents have gradually become an vital carrier or knowledge storage and information retrieval. However, traditional RAG frameworks struggle with effectively chunking documents to ensure the integrity of information, especially when dealing with QFS (Dang, 2006) and how to effectively retrieve all relevant knowledge points. In summary, when using long documents as knowledge bases in general domains, several major issues arise:(1)Naive Chunking methods are highly destructive to knowledge points (Dong et al., 2023); (2)Knowledge points become difficult to retrieve once their integrity of information is compromised (Dong et al., 2023); (3)The association between relevant knowledge points is disrupted due to suboptimal vector distances, leading to difficulties in finding all the correct knowledge points for QFS.

In recent years, advanced retrieval frameworks have emerged one after another. For instance, **Late-Chunking** (Günther et al., 2024) has proposed a "embedding then chunking" approach that cleverly generates embeddings for each text chunk that consider the entire text. **Meta-Chunking** (Zhao et al., 2024), on the other hand, introduces the concepts of Margin Sampling Chunking and Perplexity Chunking to the segmentation of text chunks, making the length of the chunks more flexible and coherent. However, the aforementioned frameworks fail to effectively exert their performance when confronted with long documents. To address this situation, Sarthi et al. (2024) proposed

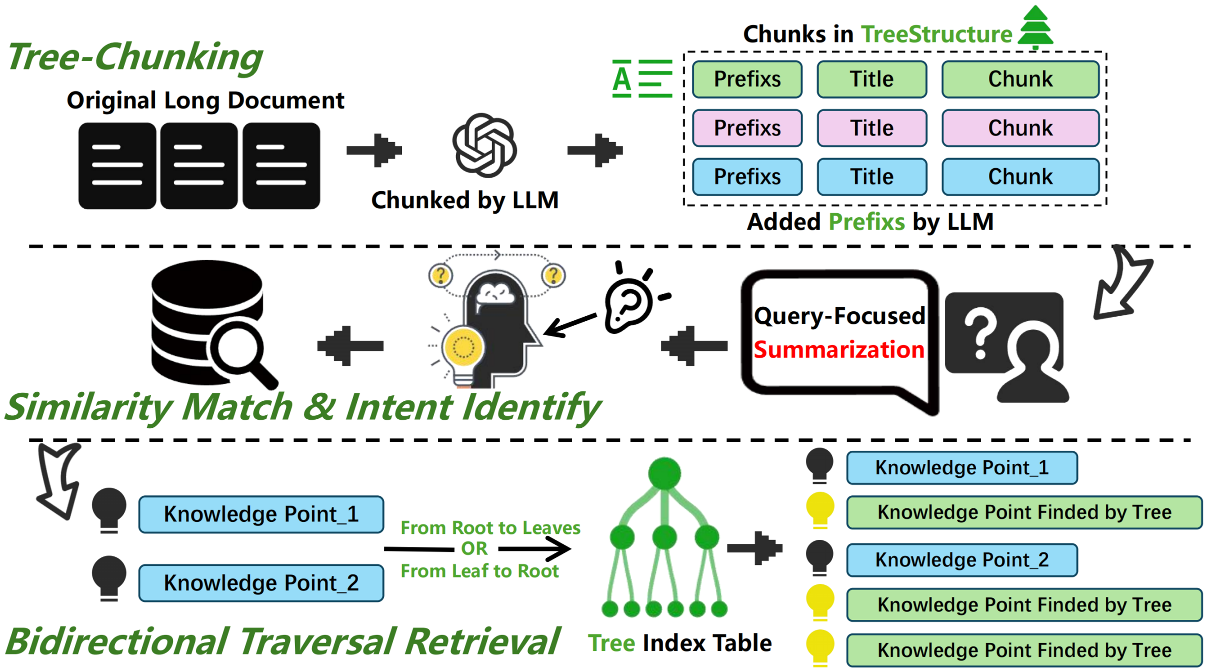

Figure 1: Framework of TreeRAG

the **RAPTOR** frameworks, which treats text chunks as nodes and constructs a tree structure from the bottom up using soft clustering to strengthen the connections between different text chunks within long documents. Nevertheless, when the subject words in the text chunks are ambiguous, the bottom-up summarization may lead to erroneous clustering issues due to the lack of clear subject. **GraphRAG** designed by Edge et al. (2024) enhances the association between information by constructing a graph structure of chunks, integrating the retrieved entities with thier related content as context input to the LLM. However, overly lengthy content may introduce excessive noise, causing the LLM to "lost in the middle (Liu et al., 2023; Yan et al., 2024; Shi et al., 2023)."

To address the aforementioned issues, in this paper, we propose a novel RAG framework called **TreeRAG**, which comprises two components: the chunking method dubbed **Tree-Chunking** and the retrieve strategy termed **Bidirectional Traversal Retrieval**.

The **Tree-Chunking** method employs a LLM to process the original documents, analyzing the general-to-specific structure within the documents in a tree-like fashion. While maintaining semantic coherence, this structure is used to hierarchically categorize the entire document, adding subtitles and index numbers. A corresponding index table dictionary is also generated for subsequent vector storage and integration with the **Bidirectional Traversal Retrieval**. When performing vector embedding of knowledge points, the original text chunk obtains the title of its immediate higher level based on its unique index number and concatenate it as a prefix. This method has been proven to effectively enhance semantic similarity (Liu et al., 2021; Karpukhin et al., 2020; Thakur et al., 2021). The rewritten text chunk is then used as the knowledge point embedding, with the original text chunk and index number serving as the metadata.

Before utilizing the **Bidirectional Traversal Retrieval**, we first employ a LLM with strong comprehension capabilities, such as GPT-4o (OpenAI et al., 2024), Qwen-max (Bai et al., 2023), Gemini (Team et al., 2024),GLM4 (Du et al., 2022) and so on, to perform a "step-back" (Zheng et al., 2024) analysis on the user's input query. It only needs to identify whether the query contains intents like summarization or concept enumeration, and based on this, decide whether to adopt this specialized retrieve strategy. Within this procession, we extract the index numbers of the TopK retrieved knowledge points then use the hierarchical positions in the tree-like index table to extract the content of their peer leaf nodes or all their subordinate

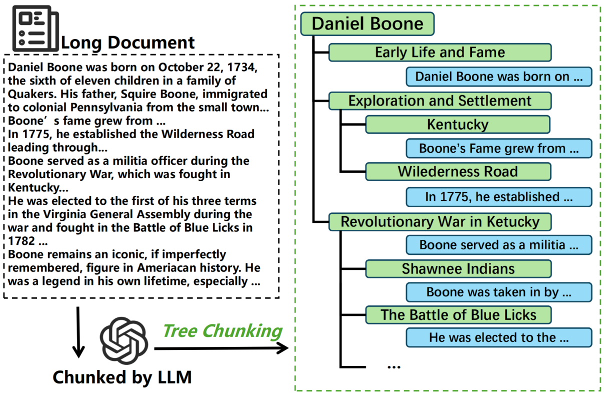

Figure 2: Chunking Example

leaf nodes. Finally, we rerank all the knowledge points to serve as the final retrieved results.

To demonstrate the reliability and underlying principles of **Tree-Chunking** and the effectiveness of the **TreeRAG** framework, we conduct ablation and comparative experiments on the **Dragonball dataset** (Zhu et al., 2024). The results show that **Tree-Chunking** effectively preserves information's integrity and connectivity in long documents, while the **TreeRAG** framework achieves good recall and generation performance with minimal noise.

Main contributions of this paper are as follows:

(1)We propose a novel text chunking method called **Tree-Chunking**, which chunks and stores text in a tree-like structure, thereby reducing the information disruption caused by chunking, and enhancing the retrieval effectiveness by completing hierarchical prefixes.

(2)We design a retrieve strategy named **Bidirectional Traversal Retrieval**, which adopts the philosophy of "**from root to leafs**" and "**from leaf to root**" to comprehensively identify knowledge points in search results, addressing to a certain extent the challenge of relevant knowledge points being distant in vector space.

(3)Experiments conducted on the **Finance, Medical and Law subsets of Dragonball dataset** demonstrate that **TreeRAG**, compared to other frameworks, has better recall quality, achieving a good recall rate while minimizing the introduction of noise.

2 Related Work

As the number of parameters and the volume of training data for LLM increase, these models have demonstrated unprecedented capabilities in handing complex language understanding and generation tasks. However, for domain-specific knowledge-intensive tasks such as open-domain question answering and fact verification, LLM still face challenges in terms of professionalism and accuracy. Consequently, RAG has emerged, combining the generative capabilities of large-scale pre-trained models with the retrieval capabilities to retrieve relevant information from a vast array of documents to assist in generation tasks.Current RAG research primarily focuses on three core stages (Gao et al., 2024) :"Retrieval," "Generation," and "Augmentation." During the retrieval stage, original documents are processed and chunked into sizes, then stored in vector databases through embedding models, and knowledge points are obtained by calculating the similarity between users' queries and document chunks in the knowledge base. In the generation stage, the retrieved knowledge points are passed to the model as contexts to assist in generating responses. The augmentation stage involves optimizing the retrieval workflow to address more complex problems. This paper focuses on the "Retrieval" and "Augmentation" stages.

Langchain[1] (Chase, 2024) offers various convenient traditional chunking strategies, such as RecursiveCharacterTextSplitter and CharacterTextSplitter. While these text splitters have their applicability in certain scenarios, they are no longer effective in meeting the increasing demand for precise knowledge recall. Particularly in long documents, a rough chunking method implies more information loss, more noise and poorer retrieval outcomes (Xu et al., 2023).

To address the aforementioned challenges, advanced frameworks have emerged. **Late-Chunking** employs chunking on documents after embedding and before mean pooling, allowing the resulting chunks to capture complete contextual information. **Meta-Chunking** introduces two chunking methods: one that identifies potential splitting points through perplexity and another that involves LLMs in sentence chunking decisions. The **RAPTOR** framework uses Uniform Manifold Approximation and Projection(UMAP) (McInnes et al., 2020) and Gaussian clustering (Bishop, 2006) to gen-

[1] https://www.langchain.com/

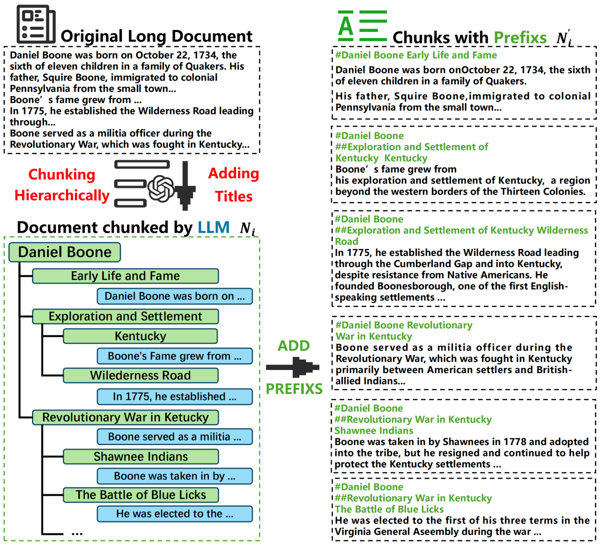

Figure 3: Prefix Add

erate nodes from the bottom up through summary generation, thereby enhancing retrieval effectiveness. **GraphRAG** optimizes final generation quality by integrating data into graph structures.

However, when facing long document knowledge bases, the challenge of effective retrieval remains. This paper argues that greater focus should be places on the connectivity between knowledge points and the preservation of hierarchical contextual information. Therefore, we propose a RAG framework called **TreeRAG** which consists of the chunking method named **Tree-Chunking** and the retrieve strategy termed **Bidirectional Traversal Retrieval** , which is designed to address these issues and enhance the performance of RAG.

3 TreeRAG

In this section, we will elaborate on the chunking method of Tree-Chunking and the construction of the index table, that is, how it chunks the text and enhances its embedding form for better retrieval effectiveness. **Bidirectional Traversal Retrieval**, based on Tree-Chunking, incorporates the use of LLM for intent identification of users' queries and the use of a tree-shaped index table for node completion. The framework of **TreeRAG** is shown in Figure 1.

3.1 Tree-Chunking

Tree-Chunking focuses on the "Retrieval" and "Augmentation" stages of RAG, consisting of two major components: the chunking method and the index table. These two components work in tandem to generate text chunks with more distinct and complete semantic features and stronger associations, as well as to create a tree-shaped index table for subsequent use in the **Bidirectional Traversal Retrieval**.

3.1.1 Chunking Method & Index Construction

Traditional chunking methods and text embedding often chunk the text based on a fixed size, and after adding a certain context window, they directly embed the chunks into local knowledge base. More advanced chunking methods that have recently emerged, such as **Late-Chunking** and **Meta-Chunking**, aim to preserve the text's association with the original document by adopting a "embedding first, then chunking" approach or by finding potential splitting points. However, their effectiveness declines as the length of the document increases. Therefore, the chunking method and embedding used in **Tree-Chunking** focus on explicitly demonstrating the relationship between chunk and its preceding text.

After performing a certain level of cleaning on the original document, an LLM with strong comprehension capabilities, like GPT-4o, is used to hierarchically categorize and add titles to the document while respecting semantic coherence and the original document's structure. The titles consist of a title index number and title content. These index numbers, generated based on the document's hierarchy, naturally form a tree-like structure from top to bottom. We represent the newly obtained chunk as N_i, which is composed of the original chunk content and the title. The original chunk content is represented as $R(N_i)$, the index number in the title is represented as $T(N_i)$, and the title content within the title is represented as $C(T(N_i))$. An example of chunking is shown in Figure 2 .

This chunking strategy flexibly divides the original document into appropriately sized and coherent text chunks, rather than using a fixed-size chunking method. To explicitly demonstrate the connection between each text chunk and the higher levels of the document, this study firstly constructs a tree-shaped index table based on the N_i. The connections and levels between nodes are determined by the title index number in the new chunk, and the content of the nodes is the original content of the new chunk. We represent this index table as D. Through this index table D, we can clearly obtain the higher-level index numbers for each title index number. Then we add the title contents within the higher-level index numbers as prefixes to N_i to enhance the accuracy of similarity retrieval. The prefix $P\left(N_i\right)$ is determined by the following formula:

$$P(N_i) = \bigcup_{i=1}^{l-1} C(T_i(N_i)), \tag{1}$$

where $\bigcup$ represents concatenation, l represents the level of the title index number, and $C(T_i(N_i))$ represents the i-th level title index number of $T(N_i)$.

The prefix $P\left(N_i\right)$ is merged with N_i to yield $N_i{}'$. This augmented chunk $N_i{}'$ is then subjected to vector embedding as a knowledge points, with the corresponding title index $T(N_i)$ and the original chunk content $R\left(N_i\right)$ being utilized as metadata. The procedure for concatenating the knowledge points is depicted in Figure 3.

3.1.2 Approaching for solving demonstrative pronoun

One of the original intentions of **Late-Chunking** is to address the ambiguity of referents for pronouns such as"It," "He," and "She" within sentences through a clever chunking method. Tree-Chunking, on the other hand, explicitly incorporates preceding text information as a prefix, which also alleviates to the situation where demonstrative pronouns and their corresponding antecedents are too far apart in the document to be understood by LLMs. A detailed comparison and experiments will be presented in the "**Experiment & Analysis**" section.

3.2 Bidirectional Traversal Retrieval

Facing QFS, such as "*Please list the effects of a certain medication*", for embedding models that have not undergone fine-tuning and have not added special tokens, the multiple concepts

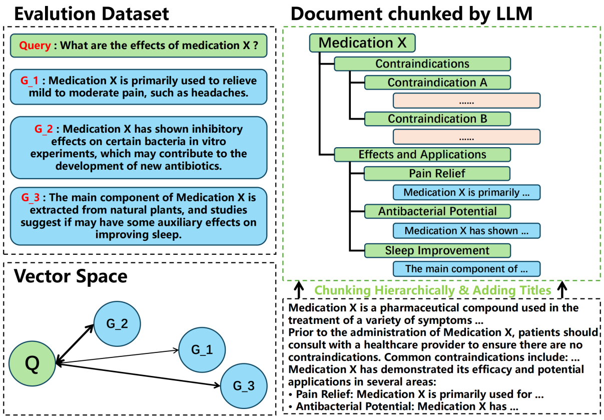

Figure 4: Unsatisfactory Vector Distance

describing the same entity may not be ideally distant from the user's query in terms of vector space, leading to the inability to fully retrieve the correct knowledge points in the ground truths. As illustrated in Figure 4, **Dataset** consists of user's queries (**Query**) and the correct knowledge points (**Ground Truths**). The Ground Truths is composed of several chunks from the **Knowledge Base** that can answer the **Query**. In the example, G_1,G_2,G_3 are all correct knowledge points for the **Query**, presenting a parallel relationship at the document hierarchy and belonging to the same node. However, in the vector space, G_2 is close to Q in terms of vector distance, while G_2 and G_3 are not ideal. Therefore, during the retrieval process, only G_2 may be included in the TopK retrieval results.

Therefore, we propose **Bidirectional Traversal Retrieval**, which utilizes LLM with strong comprehension capabilities to perform intent recognition on users' queries before retrieval. It identifies whether the queries contain concept-listing intentions such as "**Summarization**," as in the query "*What are the symptoms of disease A ?*" This query includes an intent to summarize and requires retrieving multiple knowledge points. If such an intention is detected, the process enters this special retrieve strategy; otherwise, it proceeds with the normal retrieval process. In Algorithm 1, we show this retrieval strategy. Here, T refers to the Knowledge Tree Index Table derived from Tree-Chunking, R represents the initial set of retrieved knowledge points. $T_{leaves}\left(R_i\right)$ refers to the process of obtaining all the leaf node contents associated with

R_i, while $T_{root}(R_i)$ refers to the process of extracting the unique immediate root node of R_i.

Algorithm 1: Bidirectional Traversal Retrieval

Input: Query Q; Knowledge Tree Index Table T; LLM with strong comprehension LLM$(\cdot)$; initially retrieved knowledge points R

Output: Final knowledge points F

$I \leftarrow \text{LLM}(Q)$

$F \leftarrow \emptyset$

`// Initialize` F `as an empty set`

if $I = 0$ **then**

    $F \leftarrow R$

else

    for $i = 1, 2, \ldots$ **do**

        if R_i *is a root node* **then**

            $F \leftarrow F \cup T_{\text{leaves}}(R_i)$

            `// Union with leaf nodes`

        else

            $F \leftarrow F \cup T_{\text{leaves}}(T_{\text{root}}(R_i))$

        end if

    end for

end if

Description: The LLM$(\cdot)$ determines whether the input involves a "summarization" intent. If true, it outputs 1; otherwise, it outputs 0.

Within **Bidirectional Traversal Retrieval**, the core concepts of "**From Leaf to Root**" and "**From Root to Leaves**" enable the system to retrieve all relevant knowledge points even in extreme cases where only one of the corresponding ground truths is initially retrieved. This is achieved through the relationships between root and leaf nodes. Finally, all retrieved knowledge points are re-ranked to further enhance the recall performance.

4 Experiments & Analysis

We measure **TreeRAG**'s performance on the **Dragonball dataset** (Zhu et al., 2024) through three major experiments in this section: **The Principle of Tree-Chunking**, **Comparative Experiments** and **Ablation Studies**.

The **Dragonball dataset** is a multilingual and multi-domain dataset consisting of multi-hop reasoning questions, summary questions, factual questions, and corresponding long original documents from the domains of Finance, Medical and Law. This dataset does not contain any real-world information. For more details, please refer to A.1. We select parts of dataset that contains Chinese non-multi-document questions, and in all three experiments, we use BGE-M3 (Chen et al., 2023) as embedding model which performs excellently on Chinese language tasks and utilize bold and underline formatting to indicate the highest and second-highest scores. Additionally, all pre-trained models used in experiments employ the default parameter settings.

In the **The Principle of Tree-Chunking** experiment, we use similarity as the evaluation metirc. In the **Comparative and Ablation Studies**, we use **Recall** (Musgrave et al., 2020), **Precision** and **Effective Information Rate(EIR)** Zhu et al. (2024) as metrics for retrieval quality, and **ROUGE-L** (Lin, 2004), **METEOR** (Banerjee and Lavie, 2005) and **BLEU** (Papineni et al., 2002) for generation quality evaluation.

4.1 The Principle of Tree-Chunking

In the experiments of this subsection, we will demonstrate that the method of adding explicit prefixes adopted by **Tree-Chunking** can alleviate the confusion of demonstrative pronouns, thereby proving the reliability of **Tree-Chunking** in terms of preserving the integrity and connectivity of information. We selected two long documents from the **Dragonball dataset** and extracted a coherent segment from each of them. The characteristic of each segments is that only the first sentence contains an explicit subject, while subsequent sentences use demonstrative pronouns such as "it" and "the company" to refer to that subject. To conduct a comparative experiment, this subsection will evaluate three different approaches: **Baseline**, **Late-Chunking**, and **Tree-Chunking**.

The metric for the experimental results is the cosine similarity (Zhang et al., 2020) between the subject in the first sentence of the document and each sentence in the vector space. The experimental results are presented in Table 4 and Table 5 in A.2.

In the experiments presented in Table 4, from the perspective of similarity scores, both **Late-Chunking** (Günther et al., 2024) and **Tree-**

Methods	Finance			Medical			Law		
	Recall	Precision	EIR	Recall	Precision	EIR	Recall	Precision	EIR
Late-Chunking	0.541	0.249	0.440	0.087	0.061	0.145	0.024	0.016	0.266
RAPTOR-GLM4-flashx	0.837	0.3833	0.4863	0.132	0.143	0.578	/	/	/
RAPTOR-GLM4-airx	0.835	0.382	0.492	0.119	0.150	0.540	/	/	/
Meta-Chuking-Margin	0.833	0.460	0.493	0.503	0.256	0.233	0.646	0.456	0.391
Meta-Chuking-PPL	1.513	0.609	0.321	0.594	0.325	0.171	**1.331**	**0.639**	0.298
TreeRAG	**1.983**	**0.888**	**0.630**	**0.669**	**0.415**	**1.183**	1.078	0.575	**0.807**

Table 1: **Comparative Experiment on Retrieval Quality**: The RAPTOR framework uses two different LLMs from the GLM4 series for summarizing nodes in its internal preocess.However, due to the presence of sensitive or unsafe content in the original documents of the Law subset, LLMs cannot be used for summarization. The Meta-Chunking framework, offers two different chunking logics: Margin Sampling Chunking and Perplexity Chunking.

Methods	TreeRAG	nano-GraphRAG	TreeRAG	nano-GraphRAG
	Finance		Medical	
ROUGE-L	**0.313**	0.255	0.238	**0.241**
METEOR	**0.405**	0.321	**0.319**	0.301
BLEU-1	**0.253**	0.131	**0.171**	0.101
BELU-2	**0.200**	0.106	**0.129**	0.081
BLEU-3	**0.162**	0.086	**0.105**	0.067
BLEU-4	**0.134**	0.070	**0.089**	0.056

Table 2: **Comparative Experiment on Generation Quality:** Due to the presence of unsafe and sensitive content in the Law subset, we conduct experiments on **Finance** and **Medical subsets**.

Chunking have yielded promising results.

The experiments shown in Table 5, which differ from those in Table 4 by featuring a greater number of sentences and longer sentence lengths, the superiority of **Tree-Chunking** becomes more apparent. This also theoretically demonstrates the reliability of **Tree-Chunking** in preserving the integrity and connectivity of information.

4.2 Ablation Studies & Comparative Experiments

To evaluate the performance of **TreeRAG** in addressing these challenges, we select the processed **Dragonball dataset** (Zhu et al., 2024) for our experiments, conducting tests across its **Finance, Law, Medical subsets**.

4.2.1 Comparative Experiment on Retrieval Quality

We compare **TreeRAG** with popular recall-focused RAG frameworks such as **Late-Chunking**, **Meta-Chunking** and **RAPTOR**. Among these frameworks, **Late-Chunking** and **Meta-Chunking** enhance embedding effectiveness through optimizations in the chunking method, while **RAPTOR** improves the storage structure and retrieval strategy. **TreeRAG** innovates across chunking method, storage structure and retrieval strategy to achieve better retrieval performance. The experimental results are shown in Table 1 and original results is shown in Table 6 in A.4. The final metric scores are calculated using the following formula:

$$Metric = Metric@3 + Metric@5 + Metric@10 \tag{2}$$

From this perspective reveals that **TreeRAG**, while always maintaining a great recall rate, achieves the best precision and EIR metrics, meaning it maintains the integrity and connectivity of information to the great extent while introducing the least amount of noise.

4.2.2 Comparative Experiment on Generation Quality

GraphRAG stores knowledge points in the form of a knowledge graph, integrating the various attributes the retrieved entities and presenting them to the LLM, there by enabling high-quality answer generation for QFS tasks. To ensure a fair comparison of answer generation quality across different frameworks, we choose **nano-GraphRAG** (gusye1234, 2024), which enhances the customizability of GraphRAG and is configured for Chinese QA tasks. In this experiment, we use Qwen-max as the generation model. For TreeRAG, we use the retrieved knowledge points, augmented with prefixes, as the context input to the LLM. We use **ROUGE-L**, **METEOR** and **BLEU** on **Finance** and **Medical subsets** to evaluate

	Dragonball-Finance			Dragonball-Medical			Dragonball-Law		
Method	Baseline	Tree-Chunking	TreeRAG	Baseline	Tree-Chunking	TreeRAG	Baseline	Tree-Chunking	TreeRAG
Top-3									
Recall	30.49%	47.75%	**50.51%**	1.56%	7.38%	**14.30%**	6.98%	7.64%	**26.22%**
Precision	23.11%	36.59%	**38.65%**	2.53%	10.79%	**15.38%**	10.17%	14.03%	**22.67%**
EIR	26.18%	27.67%	**27.96%**	26.97%	38.76%	**54.48%**	20.39%	24.97%	**38.48%**
Top-5									
Recall	40.09%	60.65%	**64.14%**	2.13%	9.11%	**19.55%**	12.32%	11.07%	**33.79%**
Precision	19.10%	28.33%	**29.99%**	2.08%	8.82%	**13.45%**	9.12%	11.77%	**19.53%**
EIR	18.20%	19.92%	**20.98%**	21.39%	24.05%	**38.78%**	15.46%	18.93%	**25.59%**
Top-10									
Recall	53.41%	79.82%	**83.63%**	2.65%	14.13%	**33.05%**	26.35%	19.10%	**47.76%**
Precision	12.34%	19.10%	**20.11%**	1.37%	6.62%	**12.66%**	8.46%	9.84%	**15.27%**
EIR	10.90%	13.18%	**14.04%**	11.99%	17.41%	**25.08%**	10.09%	10.88%	**16.60%**

Table 3: **Ablation Studies:** The Baseline in the study uses the same chunking method as **Tree-Chunking**, but it lacks the prefix addition step, instead opting to include a context window as a substitute. In the metrics, **EIR** quantifies the proportion of relevant information within the retrieved passages, ensuring that the retrieval process is both accurate and efficient in terms of information content.

the generation quality of **nano-GraphRAG** and **TreeRAG**. The experimental results are shown in Table 2.

The results show that TreeRAG achieves better comprehensive results, demonstrating its ability to introduce minimal noise while accurately recalling relevant knowledge points in QFS tasks, ultimately improving the quality of the LLM's answers.

4.2.3 Ablation Studies

TreeRAG is formed based on **Tree-Chunking** with the addition of a special retrieval strategy called **Bidirectional Traversal Retrieval**. To validate the effectiveness of each component within this framework, this subsection conducts ablation studies by evaluating **Baseline**, **Tree-Chunking**, and **TreeRAG** on **Dragonball dataset**.

Table 3 presents the final results of the ablation studies. The introduction of **Tree-Chunking** has yielded a noticeable enhancement in the metrics, offering a more intuitive demonstration of this chunking method's reliability. Importantly, as the components of the framework are refined step by step, there is a pronounced upward trend in the **Recall@k**. However, it is noteworthy that during this process, neither **Precision@k** nor **EIR@k** decrease as result of the framework modifications. This means that **TreeRAG** not only enhances the recall rate but also further reduces the introduction of noise. This capability sufficiently demonstrates the effectiveness of **TreeRAG** and its components in preserving the integrity and connectivity of information when addressing QFS tasks.

5 Conclusion

In this paper, we propose a tree-like structure for chunking and embedding called **Tree-Chunking**. Building upon this foundation, we introduce a RAG framework named **TreeRAG** that integrates **Bidirectional Traversal Retrieval** with the concepts of "from root to leaves" and "from leaf to root". We conduct experiments across **Dragonball dataset** to demonstrate the principle of **Tree-Chunking** in preserving the integrity and connectivity of information, thereby validating its reliability in this regard. Most importantly, we have demonstrated that **TreeRAG** can maintain the integrity and connectivity of knowledge points when tackling the QFS task on long documents, achieving high recall rates whit minimal noise introduction and ultimately facilitating the generation of high-quality answers.

Additionally, it is independent of specific embedding models and LLMs, and does not require additional training, making it applicable to a wide rage of application scenarios.

6 Limitation

In fact, during our research, we identified limitations: **TreeRAG** does not have a particular advantage when it comes to recalling knowledge points from different documents due to the independence of each constructed tree. Moreover, we have not yet focused on further optimizing the retrieved knowledge points before using them as context for input like **GraphRAG**. In the future, we plan to improve the framework's versatility and enhance its performance in QA tasks by focusing on "knowledge aggregation" and "generation enhancement".

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. *Preprint*, arXiv:2112.04426.

Harrison Chase. 2024. Langchain: A framework for developing applications powered by language models. Accessed: 2024-12-13.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

Hoa Trang Dang. 2006. Duc 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55.

Zican Dong, Tianyi Tang, Lunyi Li, and Wayne Xin Zhao. 2023. A survey on long text modeling with transformers. *Preprint*, arXiv:2302.14502.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. *Preprint*, arXiv:2103.10360.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

gusye1234. 2024. nano-graphrag: A simple, easy-to-hack GraphRAG implementation. `https://github.com/gusye1234/nano-graphrag`.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models. *Preprint*, arXiv:2409.04701.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. *Preprint*, arXiv:2007.01282.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip S. Yu. 2021. Dense hierarchical retrieval for open-domain question answering. *Preprint*, arXiv:2110.15439.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. *Preprint*, arXiv:2003.08505.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, and Haiming Bao etc. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *Preprint*, arXiv:2401.18059.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. *Preprint*, arXiv:2302.00093.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, and Heng-Tze Cheng etc. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

Shicheng Xu, Liang Pang, Huawei Shen, and Xueqi Cheng. 2023. Berm: Training the balanced and extractable representation for matching to improve generalization ability of dense retrieval. *Preprint*, arXiv:2305.11052.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *Preprint*, arXiv:2401.15884.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Jihao Zhao, Zhiyuan Ji, Yuchen Feng, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu Li. 2024. Meta-chunking: Learning efficient text segmentation via logical perception. *Preprint*, arXiv:2410.12788.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. *Preprint*, arXiv:2310.06117.

Kunlun Zhu, Yifan Luo, Dingling Xu, Ruobing Wang, Shi Yu, Shuo Wang, Yukun Yan, Zhenghao Liu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Rageval: Scenario specific rag evaluation dataset generation framework. *Preprint*, arXiv:2408.01262.

A Appendix

A.1 schema examples of Dragonball dataset

Document

Advanced Manufacturing Solutions Inc., established in August 15, 2005 in Cityville, Techland, is a publicly listed manufacturing company specializing in the design, development, and manufacturing of high-tech precision tools and equipment for various industries.
In 2017, Advanced Manufacturing Solutions Inc. underwent significant financial developments and strategic initiatives that propelled the company towards growth and success. The year started with an extensive asset restructuring process, aimed at consolidating operations and improving operational efficiency.
In March 2017, the company made a major investment of $50 million in Project Automate, a groundbreaking automated manufacturing technology. This investment expanded the company's business areas, strengthened its market position, and enhanced profitability.
...[693 words]...
This committee oversees ethical standards and ensures compliance with industry regulations. Furthermore, in October 2017, the company published its Sustainability Report, showcasing its dedication to sustainable practices and attracting socially conscious investors. The publication of this report not only demonstrates transparency but also highlights the company's efforts to minimize its environmental impact and contribute to the well-being of society. These sub-events collectively contribute to Advanced Manufacturing Solutions Inc.'s overall corporate governance structure, positioning the company as a responsible and trustworthy entity in the manufacturing industry.

Dataset

```
{"query":
{ ... ,
"query_type": "Summary Question",
"content": "Based on the corporate governance report, summarize the key corporate governance improvements made by Advanced Manufacturing Solutions Inc. in 2017."},
"ground_truth":
{"doc_ids": [42],
 "content": "In 2017, Advanced Manufacturing Solutions Inc. made several key improvements to its corporate governance. In January, the company revised its corporate governance policies to enhance transparency, accountability, and stakeholder engagement. This was ...",
 "references": ["In January 2017, Advanced Manufacturing Solutions Inc. underwent....",
"Firstly, the company successfully completed ...", "This move ...",
"Additionally, ...", ...],
 ...
}}
```

Figure 5: a schema example of Finance subsets

Document

```
**JUDGMENT**
**The People of Glenwood vs. Y. Nelson**
**1. Court and Prosecutor Information:**
*Court:* Glenwood, Quailwood Court
*Prosecutor:* Glenwood, Quailwood Procuratorate
*Chief Judge:* Hon. H. Ruiz
*Presiding Judge:* Hon. E. Collins
*Court Clerk:* K. Kelly
**2. Defendant and Defense Lawyer Information:**
*Defendant:* Y. Nelson
*Gender:* Female
*Birthdate:* December 5, 1981
...[906 words]...
(c) Forcibly taking or arbitrarily destroying or occupying public or private property, with
serious circumstances;
(d) Making trouble in a public place, causing serious disorder in the public place.
If one gathers others to repeatedly commit the aforementioned acts, seriously disrupting
social order, they shall be sentenced to fixed-term imprisonment of more than five years but
not more than ten years, and may also be fined.
```

Dataset

```
{"query":
{ ... ,
"query_type": "Summary Question",
 "content": "According to the judgment of Glenwood, Quailwood, Court, summarize the evidence
of Y. Nelson's crimes."},
 "ground_truth":
{"doc_ids": [139],
 "content": "The evidence includes multiple witness testimonies from cafe owners and market
vendors ...", "references": ["*Witness Testimony:*", "Numerous cafe owners and market
vendors testified that Y. Nelson ...", "Through her aggressive language and actions, she
caused ...", "*Surveillance Footage:*", "Security cameras in the central market and various
cafes captured Y. Nelson engaging ...", ...],
 ...
}}
```

Figure 6: a schema example of Law subsets

Document

```
**Hospitalization Record**
**Basic Information:**
Name: J. Reyes
Gender: Male
...[168 words]...
**Past History:**
General Health Condition: Generally healthy with no chronic conditions.
Disease History: No previous history of rheumatic diseases or chronic illnesses.
Infectious Disease History: No significant infectious diseases.
Immunization History: Up-to-date with routine immunizations.
Surgery and Trauma History: Appendectomy at age 30, no significant traumas reported.
...[527 words]...
**Blood Transfusion Consent:**
N/A
**Special Examination Consent:**
Consent obtained for MRI and X-rays.
**Critical Condition Notice:**
N/A
```

Dataset

```
{"query":
{ ... ,
"query_type": "Summarization Question",
 "content": "According to the hospitalization records of Bridgewater General Hospital,
summarize the present illness of J. Reyes."},
"ground_truth":
{"doc_ids": [212],
 "content": "The symptoms began insidiously 6 months ago, initially noticed while working at
a construction site. The main symptoms include morning stiffness, arthritis affecting hands,
feet, wrists, ankles, and temporomandibular joints, with pain characterized as dull and
persistent, ...",
 "references": ["Onset: The symptoms began insidiously 6 months ago, ...", "Gradual onset
with ...", "Main Symptoms: Morning stiffness, arthritis affecting ...", "Pain characterized
as ....", ...],
 ...
}}
```

Figure 7: a schema example of Medical subsets

A.2 The principle of Tree-Chunking

Text	Sim.Baseline	Sim.Late-Chunking	Sim.TreeStructure
In terms of governance structure, during the reporting period, **TuoYuan Technology Co., Ltd.** experienced several ethical and integrity issues.	**0.8206**	0.7615	<u>0.8077</u>
First, the company revealed an internal fraud case involving senior executives, who took advantage of their positions to engage in financial misconduct. This incident severely damaged the company's reputation and shareholder trust.	0.6223	**0.7393**	<u>0.7328</u>
Additionally, the company exposed issues of conflicts of interest among senior executives, including cases where executives used company resources for personal gain. These conflicts of interest further weakened the effectiveness of the company's governance.	0.6054	**0.7315**	<u>0.7164</u>

Table 4: Similarity to **TuoYuan Technology Co., Ltd.**: The "embedded-first, then-chunk" method in Late-Chunking enables each sentence's embedding vector to incorporate information from other sentences, leading to superior similarity results. In the Tree-Chunking, explicit prefixes are added to the embedded sentences, directly incorporating prior context, which also yields favorable outcomes.

Text	Sim.Baseline	Sim.Late-Chunking	Sim.TreeStructure
The year 2019 was a pivotal year in the development of **ACME R&D Co., Ltd.**, during which the company underwent a series of significant events in its financial affairs, events that had a profound impact on the company's financial status and performance.	<u>0.7600</u>	0.7288	**0.7663**
First, in June 2018, the company launched a large-scale financing plan aimed at supporting its expansion and development. After several months of preparation and negotiations, the company finalized the financing plan in September 2018 and officially signed the financing agreement in January 2019. This financing plan provided the company with sufficient funds, helping to drive its business growth and innovation in R&D.	0.6046	<u>0.6775</u>	**0.7522**
However, in March 2019, the company faced the challenge of debt restructuring. Due to the large scale of its debt, the company decided to undertake debt restructuring to reduce financial risks and ease the burden of liabilities. This measure helped to optimize the company's capital structure and improve its financial stability.	0.5454	<u>0.6728</u>	**0.7259**
In June 2019, the company made a significant investment to further expand its business scale and market share. This investment brought new growth opportunities to the company and laid a solid foundation for its future development.	0.5389	<u>0.6794</u>	**0.7444**

Table 5: Similarity to **ACME R&D Co., Ltd.**: In scenarios with extensive contents and sparse explicit subjects, although **Late-Chunking** can still perform well, the concentration of information tends to dilute as the number of sentences increases and their lengths become longer. **Tree-Chunking**, due to its explicit expression of prior context, can better maintain the association between chunks and the preceding texts, thereby offering a greater advantage in resolving demonstrative pronouns.

A.3 Retrieved Knowledge Points

Question

What is the annual budget for the future social responsibility strategy of Green Source Environmental Protection Co., Ltd.?

Knowledge Points retrieved by TreeRAG

#Green Source Environmental Protection Co., Ltd.
##Environmental Responsibility and Social Responsibility
###Future Outlook
Social Responsibility Strategy
Additionally, the company plans an annual budget of 300,000 yuan for its social responsibility strategy, actively fulfilling its social responsibilities. This includes further developing charitable activities and public welfare projects, increasing investments in employee career development and training, and enhancing cooperation and communication with the community.

#Green Source Environmental Protection Co., Ltd.
Environmental Responsibility and Social Responsibility

#Green Source Environmental Protection Co., Ltd.
##Environmental Responsibility and Social Responsibility
###Future Outlook
Environmental Protection Plan
The company plans to invest 500,000 yuan in environmental protection initiatives, taking a series of measures to reduce the environmental impact of its operations. These include continuing to promote energy-saving and emission-reduction projects, increasing research and development of green products, expanding market promotion efforts, and optimizing waste management and resource recycling.

Knowledge Points retrieved by GraphRAG

Support charitable activities and organizations in areas such as education, health, and poverty reduction. This initiative has enhanced the company's social image and brand recognition, strengthening its connections with and support from the community.According to the data, employee satisfaction in 2017 was 85%, and the community investment amounted to 10 million yuan. These figures reflect the company's efforts and achievements in social responsibility.
...[199 words]...

Green Source Environmental Protection Co., Ltd. is an environmental company established in June 2005. It is primarily engaged in the research, production, and sales of green energy, with its registered office located in Beijing. The company has not yet gone public.2017 was a key year in the development of Green Source Environmental Protection Co., Ltd. In terms of financial information, the company completed a series of important asset restructuring and investment activities, laying a solid foundation for its future development.
...[1673 words]...

Green Source Environmental Protection Co., Ltd. has carbon dioxide emissions of 10,000 tons and energy consumption of 50,000 gigajoules. The company's carbon dioxide emissions are influenced by changes in production scale and the application of energy-saving technologies, while energy consumption is impacted by improvements in production efficiency and the implementation of energy management measures. Additionally, the company's employee satisfaction rate is 80%, and community investment amounts to 200,000 yuan.
...[563 words]...

Figure 8: Knowledge Points Retrieved by RAG frameworks

A.4 Complete Results of Comparative Experiment on Retrieval Quality

Method	Late Chunking	RAPTOR GLM4-flashx	RAPTOR GLM4-airx	Meta Chunking Margin	Meta Chunking PPL	TreeRAG
Finance subset of Dragonball dataset						
			Top-3			
Recall	9.19%	20.99%	21.49%	20.64%	<u>40.66%</u>	**50.51%**
Precision	8.56%	16.74%	12.71%	20.18%	<u>27.96%</u>	**38.65%**
EIR	21.25%	23.39%	<u>24.09%</u>	23.36%	15.24%	**27.96%**
			Top-5			
Recall	15.85%	26.72%	26.68%	27.01%	<u>49.64%</u>	**64.14%**
Precision	8.75%	12.85%	16.95%	15.72%	<u>20.32%</u>	**29.99%**
EIR	14.46%	16.01%	15.84%	<u>16.48%</u>	10.62%	**20.98%**
			Top-10			
Recall	29.02%	35.99%	35.30%	35.66%	<u>61.02%</u>	**83.63%**
Precision	7.58%	8.74%	8.57%	10.11%	<u>12.58%</u>	**20.11%**
EIR	8.27%	9.23%	9.31%	<u>9.43%</u>	6.27%	**14.04%**
Medical subset of Dragonball dataset						
			Top-3			
Recall	1.61%	3.75%	3.69%	11.94%	<u>13.09%</u>	**14.30%**
Precision	2.19%	6.06%	6.33%	9.56%	<u>12.72%</u>	**15.38%**
EIR	7.36%	<u>26.49%</u>	<u>25.46%</u>	11.89%	8.18%	**54.48%**
			Top-5			
Recall	2.54%	4.95%	4.04%	15.97%	**<u>20.70%</u>**	**<u>19.55%</u>**
Precision	2.05%	4.32%	4.60%	8.11%	<u>10.56%</u>	**13.45%**
EIR	4.70%	<u>17.92%</u>	16.10%	<u>7.75%</u>	6.23%	**38.78%**
			Top-10			
Recall	4.50%	4.50%	4.21%	22.43%	<u>25.61%</u>	**33.05%**
Precision	1.84%	3.94%	4.04%	7.95%	<u>9.26%</u>	**12.66%**
EIR	2.39%	<u>13.37%</u>	12.39%	<u>3.63%</u>	2.68%	**25.08%**
Law subset of Dragonball dataset						
			Top-3			
Recall	0.07%	/	/	11.92%	**26.75%**	<u>26.22%</u>
Precision	0.17%	/	/	16.30%	**25.00%**	<u>22.67%</u>
EIR	13.91%	/	/	17.16%	<u>12.70%</u>	**38.48%**
			Top-5			
Recall	0.59%	/	/	18.77%	**42.23%**	<u>33.79%</u>
Precision	0.55%	/	/	15.59%	**22.49%**	<u>19.53%</u>
EIR	8.25%	/	/	<u>13.29%</u>	10.27%	**25.59%**
			Top-10			
Recall	1.71%	/	/	33.93	**64.16%**	<u>47.76%</u>
Precision	0.85%	/	/	13.71%	**16.41%**	<u>15.27%</u>
EIR	4.48%	/	/	<u>8.64%</u>	6.82%	**16.60%**

Table 6: Complete Results of Comparative Experiment on Retrieval Quality