Enhancing Retrieval-Augmented Generation via Evidence Tree Search

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) is widely used to enhance Large Language Models (LLMs) by grounding responses in external knowledge. However, in real-world applications, retrievers often return lengthy documents with redundant or irrelevant content, confusing downstream readers. While evidence retrieval aims to address this by extracting key information, it faces critical challenges: (1) inability to model synergistic inter-dependencies among evidence sentences, (2) lack of supervision for evaluating multi-sentence evidence quality, and (3) computational inefficiency in navigating exponentially growing search spaces of candidate evidence sets. To tackle these challenges, we propose **ETS** (Evidence Tree Search), a novel framework that reformulates evidence retrieval as a dynamic tree expansion process. Our approach first constructs an evidence tree where each path represents a candidate evidence set, explicitly modeling inter-sentence dependencies through context-aware node selection. We then leverage Monte Carlo Tree Search (MCTS) to efficiently assess evidence quality and introduce an Early-Terminating Beam Search strategy to efficiently accelerate the model inference. Extensive experiments on five datasets demonstrate that ETS significantly outperforms existing methods across different readers. Our code and datasets will be released to facilitate future research.

1 Introduction

Large Language Models (LLMs) (Taylor et al., 2022; Chowdhery et al., 2022; Zhao et al., 2023a) have demonstrated remarkable performance across a wide range of downstream tasks (Xia et al., 2024; Yamauchi et al., 2023; Imani et al., 2023; Lewkowycz et al., 2022). Despite these advancements, LLMs are still prone to generating responses that include hallucinated facts or inaccurate information (Ji et al., 2023; Shuster et al., 2021; Zhang et al., 2023a), which undermines their reliability.

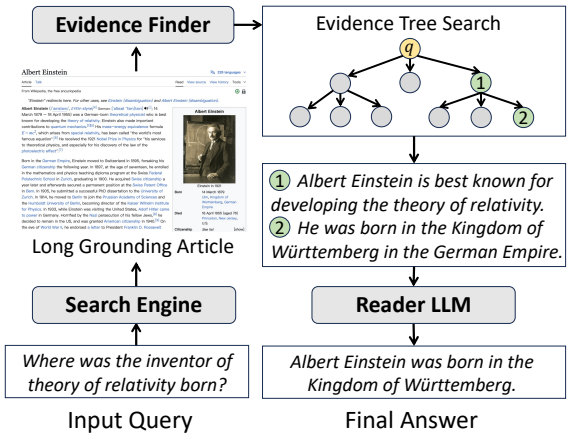


Figure 1: Illustration of Evidence Retrieval for RAG. Given an input query, a long article is retrieved using a search engine and segmented into sentences. The evidence finder then performs a reward-guided tree search to identify the optimal set of evidence, which is subsequently passed to the reader LLM for answer generation.

To mitigate this issue, Retrieval-Augmented Generation (RAG) has been introduced, which integrates external knowledge into the generation process (Ram et al., 2023; Shi et al., 2023; Rashkin et al., 2021; Gao et al., 2022; Bohnet et al., 2022; Menick et al., 2022).

In typical RAG systems, a user's query is utilized to retrieve relevant documents from external sources, which are then fed into the reader LLM to produce more accurate responses. However, in real-world applications, imperfect retrievers often return long documents, such as web pages, that may contain substantial irrelevant content. Only specific portions of these documents are relevant for answering the query. Directly using these lengthy documents in RAG systems poses several challenges: readers may struggle with processing long texts, and irrelevant content could distract the reader from generating the correct response. Therefore, evidence retrieval plays a critical role in addressing these issues by identifying and extracting the most

relevant evidence sentences from the retrieved documents before passing it to the reader.

However, evidence retrieval is a non-trivial task due to several inherent challenges. **First**, relevant evidence often comprises multiple interrelated pieces of information. Traditional retrieval methods focus on modeling the relationship between the query and each individual sentence, inherently failing to capture the complex inter-dependencies among evidence sentences. **Second**, assessing the quality of evidence sets presents significant difficulties due to the lack of explicit supervision signals. Most RAG corpora only provide binary document-level relevance labels, offering no direct feedback on which specific sentence combinations optimally support answer generation. **Finally**, the combinatorial explosion of candidate evidence sets in lengthy documents creates an enormous search space, making it computationally intensive to efficiently identify the optimal set of evidence.

To address these challenges, as depicted in Figure 1, we propose **ETS**, a novel evidence retrieval framework with **E**vidence **T**ree **S**earch. **To capture the complex inter-dependencies among evidence sentences**, ETS model evidence retrieval as a dynamic tree expansion process, where each node represents a sentence and every root-to-leaf path constitutes a candidate evidence set. This hierarchical structure inherently captures inter-sentence dependencies — during tree growth, new sentences are selected based on accumulated contextual evidence along the current path, enabling holistic evaluation of sentence combinations' collective relevance to the query. **In order to assess the quality of evidence sets**, we introduce Monte Carlo Tree Search (MCTS) (Silver et al., 2017) to guide the exploration of potential evidence combinations. Through iterative MCTS simulations, rewards derived from reader correctness given the candidate evidence set are back-propagated to update node values, providing dense training signals for optimizing evidence selection. **To combat the exponential search space**, we propose pruning the search space using a value model trained on reward signals derived from the evidence tree constructed through MCTS annotation. By leveraging this model, the framework efficiently narrows down the search space, retaining only the most promising evidence combinations. Additionally, to further reduce inference latency, we introduce Early Terminating during the expansion phase of beam search. This technique terminates the generation of candidate sentences early when a sentence prefix uniquely identifies its position in the original context, significantly improving computational efficiency without compromising accuracy.

Our contributions are summarized as follows:

- We propose ETS, a tree-based framework that enhances LLMs' evidence-grounding through a tree expansion process.
- We introduce MCTS to assess evidence quality and Early Terminating Beam Search to reduce search space and improve selection efficiency.
- Extensive experiments on five datasets show our method outperforms existing approaches across different readers.

2 Background

In Retrieval-Augmented Generation (RAG), a retriever first retrieves relevant text evidence $\mathcal{D}$ from a corpus $\mathcal{C}$ given a query q. The retrieved evidence is then used by a reader LLM to generate an answer:

$$\mathcal{D} = \text{Retriever}(q, \mathcal{C}), \quad \mathcal{A} = \text{Reader}(q, \mathcal{D}),$$

The retriever can be implemented using open-source models (e.g., DPR (Karpukhin et al., 2020)) or commercial search engines (e.g., Google), while the reader is typically a LLM.

In practice, retrievers often return lengthy documents (e.g., web pages or articles) instead of concise evidence, making it difficult for the reader to generate accurate answers. To address this, evidence retrieval aims to identify and extract a subset of key text spans, referred to as supporting evidence $\mathcal{T} = \{t_1, \ldots, t_k\} \subset \mathcal{D}$, that are most relevant to answering the query q. This selection process can be formalized as a mapping function $f(\cdot)$:

$$\mathcal{T} = \{t_1, \ldots, t_k\} = f(\mathcal{D}).$$

In this work, we define $f(\cdot)$ as a tree expansion process, where each node represents a sentence, and the path from the root node to a leaf node represents the selected evidence.

3 Approach

3.1 Overview

Figure 3 presents an overview of our framework. The process begins with the application of the

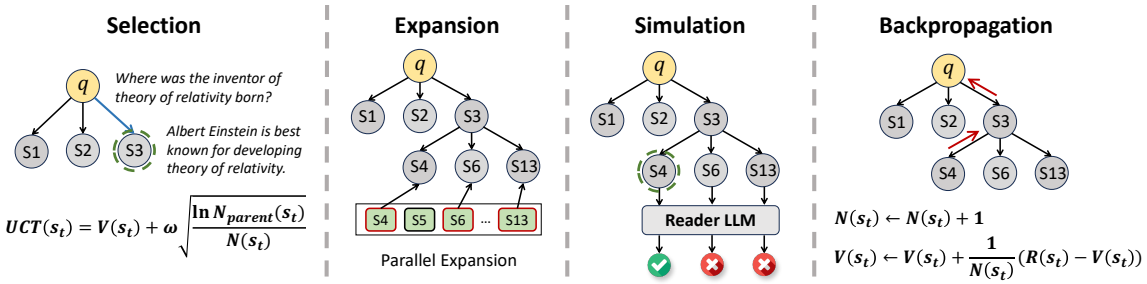


Figure 2: Single iteration of MCTS Annotation. The iteration is repeated until the maximum number of iterations is reached or no further nodes in the tree can be expanded.

Monte Carlo Tree Search (MCTS) algorithm to construct evidence trees for the queries in the training dataset. From these trees, we extract both correct and incorrect paths, which are subsequently utilized to train the policy model and the value model, respectively. During the inference phase, we introduce an early terminating beam search mechanism. This approach efficiently minimizes the search space and terminates the generation process once the prefix uniquely identifies the evidence, thereby optimizing computational efficiency and accuracy. Next, we will first illustrate the process of the MCTS annotation.

3.2 MCTS Annotation

Given an input query, a long article is retrieved using a search engine and segmented into individual sentences, which serve as the initial candidate set. However, due to the excessive length of the article, the candidate set can become prohibitively large, with many sentences being irrelevant to the query. This not only increases the computational complexity of MCTS but also introduces noise into the labeling process. To address this issue, we introduce a filtering mechanism that leverages the BGE model (Xiao et al., 2023) to compute the similarity between the query and each candidate sentence. By retaining only the most semantically similar sentences, we significantly reduce the candidate set, thereby improving the overall efficiency and accuracy of the annotation process.

During the search process, MCTS runs for multiple simulations. This process terminates when the maximum iteration number is reached or no paths can be expanded. For the i-th simulation, it conducts four operations to expand the tree:

Selection During the i-th simulation of the MCTS, the process begins with s_0, which represents the input query. The algorithm then proceeds to explore the tree by selecting nodes according to the Upper Confidence Bound for Trees (UCT) criterion (Rosin, 2011). This selection process is mathematically represented as:

$$UCT(s_t) = V(s_t) + w\sqrt{\frac{\ln N_{parent}(s_t)}{N(s_t)}} \quad (1)$$

where w controls the balance between exploration and exploitation. The node value $V(s_t)$ and its visiting count $N(s_t)$ will be updated as the search progresses. $N_{parent}(s_t)$ represents the visiting count of the parent node of s_t.

Expansion After selecting the node to be expanded, the next step is to choose the most suitable sentence from the candidate set. A straightforward way to evaluate the quality of the evidence set is to measure the likelihood of the reader generating the correct answer when the evidence set is used as the model input. To accelerate the labeling process, we employ parallel expansion, where we concurrently compute the likelihood of outputting the correct answer after incorporating each candidate sentence into the evidence set. This is expressed as:

$$R(s_t) = P(y|q, d_t) \quad (2)$$

where y represents the correct answer, and d_t denotes the evidence set after adding the candidate sentence s_t. From these calculations, we select the top M sentences that are most likely to lead to the final correct answer as the expansion nodes.

Simulation The simulation phase aims to determine whether the current path can lead the reader to generate the final correct answer. To achieve this, for each node, we construct the evidence set by concatenating the sentences from the root node to the current node. The reader then answers the question based on this evidence set. If the reader

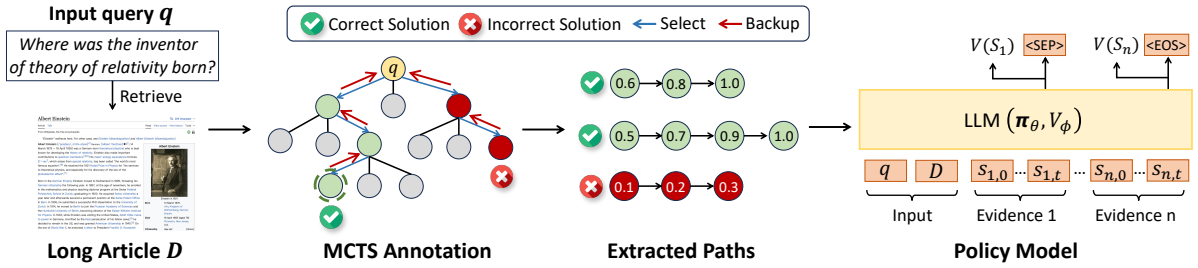


Figure 3: Overview of the proposed method: Given an input query, a long article is retrieved using a search engine and segmented into individual sentences, which serve as the initial candidate set. Then the MCTS algorithm constructs evidence trees for the input query from which correct and incorrect paths are extracted to train the policy and value models. During inference, the early terminating beam search minimizes the search space and terminates generation once the generated prefix uniquely identifies a sentence in the candidate set.

provides the correct answer, indicating that sufficient evidence has been identified, the reward is set to 1. Otherwise, the reward is defined as the likelihood of the reader outputting the correct answer based on the evidence set.

Backpropagation At the end of the i-th simulation, each edge along the path from the leaf node s_t to the root undergoes a backward pass update. The updates to their values and visiting counts are executed according to the following rule:

$$\begin{aligned} N(s_t) &\leftarrow N(s_t) + 1 \\ V(s_t) &\leftarrow V(s_t) + \frac{1}{N(s_t)}(R(s_t) - V(s_t)) \end{aligned} \tag{3}$$

where $N(s_t)$ represents the visiting count of node s_t, and $V(s_t)$ denotes its value.

3.3 Model Training

In our framework, the policy model π_θ is initialized with a pre-trained LLM. We extend this model to derive the value model V_ϕ by adding an auxiliary linear layer with a `Sigmoid` activation function. This layer operates alongside the traditional softmax layer responsible for token prediction, as illustrated in the rightmost panel of Figure 3. This design ensures that the policy model and the value model share the majority of their parameters, promoting parameter efficiency and joint optimization.

To construct the training signals for the policy model and the value model, we sample solution paths from the tree constructed through multiple rounds of MCTS. These paths are denoted as $\mathbf{x}^+$ (correct solutions) and $\mathbf{x}^-$ (incorrect solutions). We then apply a multi-task loss function to jointly update both models:

$$\mathcal{L} = -\log \pi_\theta(\mathbf{x}^+|\mathbf{q}) + \beta \cdot \sum_{t=1}^{T(\mathbf{x})} \|V_\phi(\mathbf{s}_t) - V(\mathbf{s}_t)\|^2 \tag{4}$$

Here, the first term represents the negative log-likelihood loss for next-token prediction in correct solutions, which guides the policy model to generate accurate predictions. The second term captures the loss in value prediction for both correct and incorrect solutions, ensuring the value model provides reliable estimates of expected rewards at each node. $T(\mathbf{x})$ denotes the number of steps in the solution path $\mathbf{x}$, and β is a tunable hyperparameter that controls the weight of the value loss term.

3.4 Model Inference

After obtaining the trained policy model, it can be directly used to generate grounding evidence. However, this iterative generation process has two limitations: it fails to fully explore the evidence space, and the generation can be slow due to the large input length. To address these issues, we propose an early terminating beam search mechanism, which efficiently explores the evidence space while bypassing unnecessary decoding steps.

Early Terminating Beam Search Specifically, given the input query and the long article, the policy model is used to generate multiple candidate sentences. Since the generation target originates directly from the source text, once the generation prefix is determined, the generated prefix can be used to locate the corresponding text in the source article. This allows us to skip decoding intermediate tokens, significantly speeding up the process. The value model is then employed to evaluate the quality of the expanded sentences, retaining only the most valuable nodes. This process is iterated until the maximum depth is reached or no further paths can be expanded. Finally, the answer with the highest value is selected as the output.

Model	2WikiMulti		HotpotQA		MusiQue		MultiField		Qasper		AVG	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Qwen2.5-14B-Instruct-1M												
BM25	32.00	30.08	39.50	38.32	13.50	12.09	25.33	39.90	16.00	16.30	25.27	27.34
BGE-base-en	37.50	37.01	40.50	38.45	19.50	19.98	27.33	44.07	21.50	23.74	29.27	32.65
LLM-Embedder	36.00	36.91	39.50	36.56	20.50	20.21	26.67	45.92	25.50	26.97	29.63	33.31
ChatGLM3-6B	32.50	30.90	30.00	31.59	14.00	15.03	16.00	33.61	21.50	23.14	22.80	26.85
Qwen2.5-7B	40.50	37.20	46.00	42.59	18.50	18.67	24.00	41.50	22.50	23.74	30.30	32.74
GLM-4-9B	37.50	37.25	43.50	44.28	25.00	24.85	28.67	42.05	19.00	18.94	30.73	33.47
Qwen2.5-14B	35.00	32.11	42.00	40.42	22.00	22.06	22.67	41.07	25.00	27.06	29.33	32.54
Qwen2.5-32B	42.50	36.39	43.00	41.81	18.50	19.01	19.33	32.66	15.00	15.38	27.67	29.05
Qwen2.5-72B	41.00	38.09	43.00	43.68	19.50	21.96	24.67	37.07	13.50	14.60	28.33	31.08
CFIC-7B	45.00	42.52	45.50	39.06	27.50	26.06	28.67	45.94	26.00	27.92	34.53	36.30
ETS-7B	**58.00**	**56.23**	**51.00**	**51.26**	**39.50**	**39.85**	**29.33**	**48.38**	**27.00**	**29.02**	**40.97**	**44.95**
Qwen2.5-72B-Instruct												
BM25	31.00	28.59	42.00	40.45	12.00	14.62	23.33	39.96	16.50	20.52	24.97	28.83
BGE-base-en	41.00	40.83	43.50	42.98	19.50	19.79	28.67	46.02	24.50	27.23	31.43	35.37
LLM-Embedder	40.00	40.36	41.50	41.56	21.00	21.65	27.33	45.11	27.50	30.10	31.47	35.75
ChatGLM3-6B	41.50	33.19	39.50	35.39	15.00	15.32	13.33	32.13	18.50	22.08	25.57	27.62
Qwen2.5-7B	44.00	43.25	46.50	44.50	20.50	22.09	24.00	41.14	20.50	27.26	31.10	35.65
GLM-4-9B	43.50	38.55	48.00	44.10	25.50	27.13	28.67	45.95	21.00	24.93	33.33	36.13
Qwen2.5-14B	43.00	41.21	43.00	42.42	23.00	25.42	21.33	40.19	22.00	27.10	30.47	35.27
Qwen2.5-32B	52.00	41.37	47.00	44.23	21.50	22.90	18.00	35.68	17.50	23.31	31.20	33.50
Qwen2.5-72B	53.00	38.79	47.50	42.77	23.50	25.50	22.67	39.90	15.50	21.58	32.43	33.71
CFIC-7B	54.00	43.67	53.00	48.46	30.50	31.26	29.33	46.83	29.00	30.81	39.17	40.21
ETS-7B	**60.50**	**54.82**	**60.50**	**57.72**	**40.00**	**40.84**	**30.00**	**47.87**	**31.00**	**34.91**	**44.40**	**47.23**

Table 1: Main experiment results on five QA datasets across two reader LLMs. The best results are in bold. The AVG column shows the average EM and F1 scores across all datasets.

4 Experiments

4.1 Datasets and Metrics

We conduct experiments on five datasets from LongBench (Bai et al., 2023), including 2WikiMultihopQA (Ho et al., 2020), HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), MultiFieldQA (Bai et al., 2023), and Qasper (Dasigi et al., 2021). Following the LongBench benchmark, we use the EM score and F1-score as evaluation metrics. For further details on LongBench, please refer to Bai et al. (2023). The statistical information of all datasets is provided in Table 4.

4.2 Baselines

In this study, we benchmark our method against the following three categories of baselines:

Retrieval-Based Methods Retrieval-based methods typically segment lengthy documents into smaller passages and use retrieval techniques to prioritize relevant passages. For a fair comparison, we use sentences as the retrieval unit and employ BM25 (Yang et al., 2017), BGE-base-en-v1.5 (Xiao et al., 2023), and LLM-Embedder (Zhang et al., 2023b) as retrieval models. The highest-ranking sentences are selected as input context for the reader to support QA tasks.

Extraction-Based Methods For extraction-based methods, we evaluate state-of-the-art long-context models, including ChatGLM3-6B-128K (GLM et al., 2024), Qwen2.5-7B-Instruct-1M (Team, 2025), GLM-4-9B-Chat-1M (GLM et al., 2024), Qwen2.5-14B-Instruct-1M (Team, 2025), Qwen2.5-32B-Instruct (Team, 2024), and Qwen2.5-72B-Instruct (Team, 2024). These models refine lengthy documents into concise text evidence, which serves as context for the reader LLM to support QA tasks.

Learning-Based Methods Unlike the above methods, which use off-the-shelf models, learning-based methods fine-tune models specifically for the task of evidence retrieval to improve performance. We implement the current state-of-the-art

model, CIFC (Qian et al., 2024), following its open-sourced code and data.

To ensure fairness, all models provide a comparable volume of textual evidence for the reader.

4.3 Implementation Details

During the MCTS annotation process, the Qwen2.5-7B-Instruct-1M model is utilized as the reader. The maximum number of iterations is configured to 20. Following Chen et al. (2024), the parameter w, β is set to 1.4 and 0.1, respectively. For the training dataset, we sample 6,000 correct paths and 6,000 incorrect paths. The policy model is initialized using the Qwen2.5-7B-Instruct-1M. Training involves fine-tuning the model for two epochs with a batch size of 1 and a learning rate of 1×10^{-6}, utilizing 8 NVIDIA A100 80GB GPUs. To evaluate the generalizability of our approach, we employ two additional reader models: Qwen2.5-14B-Instruct-1M and Qwen2.5-72B-Instruct. We intend to open-source the code to facilitate the reproducibility of our methodology. For more implementation details, please refer to Appendix B.

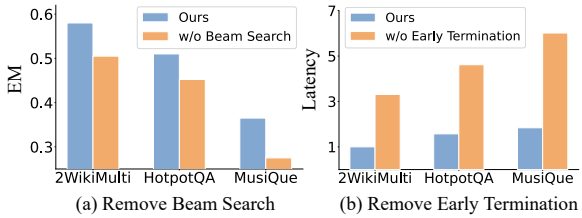
4.4 Main Results

In this section, we present the experimental results on five QA datasets using Qwen2.5-14B-Instruct-1M and Qwen2.5-72B-Instruct as the readers. Based on the results shown in Table 1, we can have the following observations:

First, our method consistently outperforms baselines across all datasets and readers, demonstrating its effectiveness. Notably, with Qwen2.5-14B-Instruct-1M as the reader, our approach achieves a 22% relative average improvement over the best-performing baseline. By modeling evidence retrieval as a tree expansion process, we enable the policy model to thoroughly explore the sentence space, significantly enhancing performance.

Second, among the baselines, smaller retrieval models such as BGE-base-en exhibit strong performance compared to extraction-based methods, aligning with findings from Qian et al. (2024). This is attributed to the retrieval method's ability to precisely compare query-sentence relevance, whereas extraction-based methods may overlook specific sentences, leading to information loss.

Third, the performance gap between Qwen2.5-14B-Instruct-1M and Qwen2.5-72B-Instruct is relatively small despite their substantial difference in parameter sizes. This underscores the effectiveness of our policy model, which alleviates the reader's burden by precisely identifying relevant evidence. As a result, users can choose smaller reader models without sacrificing model performance. This finding is particularly impactful, as it highlights the potential to substantially reduce inference costs while maintaining high-quality results.



(a) Remove Beam Search (b) Remove Early Termination

Figure 4: We conduct an ablation study by removing beam search and early termination. The latency is calculated using the latency of our method as the basic unit.

4.5 Discussion

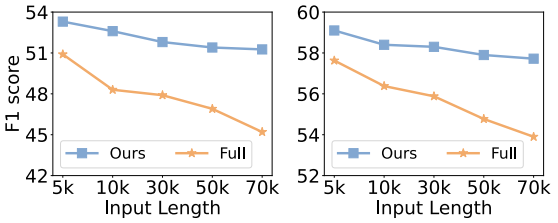
Ablation Study During inference, we propose early terminating beam search to accelerate the process while ensuring comprehensive exploration of the sentence space. To evaluate its impact, we systematically remove beam search and early termination, conducting experiments on the 2WikiMultihopQA, HotpotQA, and MusiQue datasets. The results are shown in Figure 4.

Removing beam search significantly degrades performance, as the model fails to fully explore the sentence space, leading to suboptimal evidence retrieval. In contrast, removing early termination substantially improves latency, as the model can uniquely identify the location of the original sentence early in the decoding process, eliminating the need to decode subsequent tokens. Additionally, as the input length increases across the datasets, the inference latency grows significantly. However, with early termination, the additional latency caused by longer inputs is minimal, demonstrating its effectiveness in long-input scenarios. These results highlight the importance of both components in balancing performance and efficiency.

Hyperparameter Study In our method, beam search is employed to efficiently explore the sentence space, with two key hyperparameters—Expansion Size B_1 and Beam Size B_2—playing a crucial role. To investigate their impact on model performance, we conduct experiments on the 2WikiMultihopQA, HotpotQA, and MusiQue datasets, tuning these parameters within the range of 1 to 5.

Size	HotpotQA		2WikiMulti		MusiQue	
	EM	F1	EM	F1	EM	F1
Expansion Size B_1						
1	49.00	50.72	55.00	53.98	29.00	29.63
2	50.50	50.41	60.50	56.80	35.50	34.80
3	51.00	51.26	58.00	56.23	39.50	39.85
4	55.00	55.30	59.50	58.21	39.00	38.70
5	53.00	55.28	61.00	60.52	37.00	36.74
Beam Size B_2						
1	49.00	51.76	56.00	54.21	32.50	33.33
2	50.50	51.68	61.00	58.88	35.50	36.20
3	51.00	51.26	58.00	56.23	39.50	39.85
4	51.50	52.86	61.50	59.65	32.50	33.37
5	51.00	51.76	60.50	57.91	31.00	32.37

Table 2: Hyperparameter study. We tune the parameters using Qwen2.5-14B-Instruct-1M as the reader.



(a) Qwen2.5-14B-Instruct-1M (b) Qwen2.5-72B-Instruct

Figure 5: Effect of input length. We vary the input length and observe the performance change.

Expansion size indicates the number of sentences expanded at each layer while beam size refers to the number of sentences retained at each layer during the search process. The results reveal a consistent pattern: performance peaks when the beam size or expansion size is set to approximately 4. Values smaller or larger than this threshold lead to a decline in performance. This is primarily because a larger beam size or expansion size offers the model a broader scope to explore potential sentences, increasing the likelihood of identifying the most relevant and optimal candidates. However, when these values exceed a certain threshold, the model encounters challenges in effectively ranking and selecting the best sentences from the expanded pool. This inefficiency arises because the increased search space introduces more noise and less relevant candidates, making it harder for the model to discern the best options. Additionally, increasing these values incurs higher computational costs, highlighting the need for a balanced configuration.

Effect of Input Length In this section, we analyze the effect of input length on model performance. Specifically, we vary the input length across {5000, 10000, 30000, 50000, 70000} and observe the resulting performance changes, comparing our method with using the full article as grounding evidence.

As shown in 5, both models exhibit a performance decline as input length increases. However, our method consistently outperforms the full-article input approach. While the performance of the full-article method drops significantly, our method shows only a slight decrease, demonstrating its robustness. This is because long-context models struggle to effectively capture the interaction between the query and the entire sentence set as input length grows. Our method addresses this challenge by modeling the process as node expansion, where the policy model is guided to thoroughly explore the sentence space at each step, ensuring robust performance even with longer inputs.

4.6 Case Study

In this section, we compare the evidence retrieval results of different methods using the MusiQue dataset. Based on the examples shown in Table 3, several observations can be made:

First, both the retrieval-based method BGE-base-en and the extraction-based method Qwen2.5-72B fail to accurately locate the relevant grounding text, leading to the generation of irrelevant information for downstream tasks. Although CFIC-7B can identify the location of the grounding text, it lacks the necessary background information, making it difficult for the reader to deduce the correct answer. In contrast, both GPT-4 and our method successfully identify two critical entities: Adam Smith and the University of Cambridge. However, our method distinguishes itself by offering a more detailed and comprehensive grounding text. Specifically, our approach provides a clearer explanation of the relationship between Adam Smith and the Lectures on Jurisprudence, which is crucial for understanding the context. This enhanced clarity and depth of information make it significantly easier for readers to deduce the correct answer, as the grounding text not only identifies the relevant entities but also elaborates on their connections in a way that is both informative and accessible. Given the large parameter size gap between GPT-4 and our method, we believe our approach demonstrates strong performance at a minimal computational cost.

Query: Who gives out the prize named after the author of Lectures on Jurisprudence? **Answer**: University of Cambridge
BGE-base-en: Lectures on Jurisprudence, also called Lectures on Justice, Police, Revenue and Arms (1763) is a collection of Adam Smith's lectures... The prize winner receives a monetary award and will complete a lecture tour within the UK.
Qwen2.5-72B: The prize - named after Scottish philosopher and economist Adam Smith - was originally established in 1891 and awarded triennially for the best submitted essay on a subject of the writer's choice..
CFIC-7B: "A library of his scholarly works, notes, and correspondences is held by the University of Cambridge. The Nobel Prize was not awarded between 1940 and 1942 due to the outbreak of World War II.
GPT-4: The Adam Smith Prizes are prizes currently awarded for the best overall examination performance and best dissertation in Part IIB of the Economics Tripos ... at the University of Cambridge.
ETS-7B: Lectures on Jurisprudence, also called Lectures on Justice ... is a collection of Adam Smith's lectures... The Adam Smith Prizes are prizes currently awarded for the best overall examination ... at the University of Cambridge.

Table 3: Examples of evidence retrieval on dataset MusiQue. The text colored with teal refers to the key information related to the user query.

5 Related Work

5.1 Retrieval-Augmented Generation

Despite significant progress, Large Language Models (LLMs) still produce responses that contain hallucinated facts and inaccuracies (Ji et al., 2023; Shuster et al., 2021; Zhang et al., 2023a), undermining their overall reliability. To address this issue, Retrieval-Augmented Generation (RAG) has been introduced as a method to integrate external knowledge and enhance the accuracy of model responses (Ram et al., 2023; Shi et al., 2023; Rashkin et al., 2021; Gao et al., 2022; Bohnet et al., 2022; Menick et al., 2022).

Among existing approaches, some studies propose retrieving information only once at the beginning of the generation process (Shi et al., 2023; Wang et al., 2023b; Zhang et al., 2023c; Yu et al., 2023a,c). Other works (Qian et al., 2023; Yu et al., 2023b) suggest retrieving information multiple times throughout the generation process, offering greater flexibility in determining when and what to retrieve. For example, Jiang et al. (2023) advocate for retrieval only when the generation model produces low-confidence tokens, while Ram et al. (2023) recommend refreshing retrieved documents every n tokens, a method shown to outperform single retrieval approaches. Additionally, Wang et al. (2023a); Asai et al. (2023); Zhao et al. (2023b) propose retrieving information only when the LLM determines it is necessary, further improving retrieval efficiency.

5.2 Evidence Retrieval in RAG

While RAG has demonstrated strong performance, it faces challenges when dealing with lengthy and complex retrieved documents. To address this limitation, techniques like chunking and retrieval have been developed to improve passage relevance. For instance, Mao et al. (2021) introduced a chunking and retrieval method to enhance evidence selection, while Guu et al. (2020); Lin et al. (2023) proposed jointly training the retriever and generator to improve knowledge utilization and contextual understanding.

However, chunking can often be suboptimal, as determining the appropriate granularity for chunking is a challenging task. Improper chunking can disrupt the semantic coherence of the document, leading to less accurate evidence selection. Other approaches (Ratner et al., 2022; Xu et al., 2023) focus on adapting large language models to process longer contexts by training them on extended text lengths. For example, Chen et al. (2023) extended the context size of LLMs via parameter-efficient fine-tuning. Qian et al. (2024) proposed a chunking-free method to identify relevant evidence for user queries. Similarly, FILCO (Wang et al., 2023b) filters the input context with a context-filtering model. Despite these advancements, these methods often struggle to model the complex interactions between sentences in long contexts and may fail to accurately pinpoint the grounding evidence.

6 Conclusion

In this paper, we propose ETS, a novel framework that enhances evidence retrieval in RAG systems. By modeling evidence retrieval as a tree expansion process and leveraging MCTS and Early Terminating Beam Search, ETS effectively addresses the challenges of complex evidence interdependencies, lack of supervision signals, and large search spaces. Our extensive experiments demonstrate that ETS outperforms existing methods.

Limitations

In this paper, we propose a tree expansion method for evidence retrieval. We acknowledge two limitations of our method. First, The MCTS annotation requires multiple simulations, which can result in additional labeling costs. Second, due to resource constraints, we conduct experiments using only Qwen2.5-7B-Instruct-1M as the model backbone. We leave experiments with larger model backbones for future work.

Ethics Statement

This work complies with the ACL Ethics Policy. All datasets and LLMs used are publicly available. Our research focuses on evidence retrieval, and we do not anticipate any negative ethical impacts.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *Preprint*, arXiv:2308.14508.

Bernd Bohnet, Vinh Q Tran, Pat Verga, Roee Aharoni, Daniel Andor, Livio Baldini Soares, Jacob Eisenstein, Kuzman Ganchev, Jonathan Herzig, Kai Hui, et al. 2022. Attributed question answering: Evaluation and modeling for attributed large language models. *arXiv preprint arXiv:2212.08037*.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: process supervision without process. *arXiv preprint arXiv:2405.03553*.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. *Preprint*, arXiv:2105.03011.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2022. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

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.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo

Gutman-Solo, et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Rider: Reader-guided passage reranking for open-domain question answering. *arXiv preprint arXiv:2101.00294*.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. 2022. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*.

Hongjin Qian, Zhicheng Dou, Jiejun Tan, Haonan Chen, Haoqi Gu, Ruofei Lai, Xinyu Zhang, Zhao Cao, and Ji-Rong Wen. 2023. Optimizing factual accuracy in text generation through dynamic knowledge selection. *arXiv preprint arXiv:2308.15711*.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. *arXiv preprint arXiv:2402.09760*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2022. Parallel context windows improve in-context learning of large language models. *arXiv preprint arXiv:2212.10947*.

Christopher D Rosin. 2011. Multi-armed bandits with episode context. *Annals of Mathematics and Artificial Intelligence*, 61(3):203–230.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *CoRR*, abs/2211.09085.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen2.5-1m: Deploy your own qwen with context length up to 1m tokens.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Preprint*, arXiv:2108.00573.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023a. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. 2023b. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating mathematical reasoning beyond accuracy. *arXiv preprint arXiv:2404.05692*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *arXiv preprint arXiv:2310.03025*.

Ryutaro Yamauchi, Sho Sonoda, Akiyoshi Sannai, and Wataru Kumagai. 2023. Lpml: llm-prompting markup language for mathematical reasoning. *arXiv preprint arXiv:2309.13078*.

Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of lucene for information retrieval research. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 1253–1256.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023a. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023b. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv:2305.14002*.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023c. Augmentation-adapted retriever improves generalization of language models as generic plug-in. *arXiv preprint arXiv:2305.17331*.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023a. Sac3: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency: Reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023b. Retrieve anything to augment large language models. *Preprint*, arXiv:2310.07554.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023c. Merging generated and retrieved knowledge for open-domain qa. *arXiv preprint arXiv:2310.14393*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023a. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, and Jianshu Chen. 2023b. Thrust: Adaptively propels large language models with external knowledge. *arXiv preprint arXiv:2307.10442*.

A Dataset Statistics

Settings	2WikiMultihopQA (Ho et al., 2020)	HotpotQA (Yang et al., 2018)	MuSiQue (Trivedi et al., 2022)	MultiFieldQA (Bai et al., 2023)	Qasper (Dasigi et al., 2021)
Task	Multi-doc QA	Multi-doc QA	Multi-doc QA	Single-doc QA	Single-doc QA
Train Data	5,000	5,000	5,000	0	0
Test Data	200	200	200	200	200
Average Length	29,495	56,446	69,269	28,947	23,640
Metrics	EM, F1	EM, F1	EM, F1	EM, F1	EM, F1

Table 4: Statistics and experimental settings of different tasks/datasets.

B Training Details

Dataset Construction We sample 5000 queries from the training data of 2WikiMultihopQA, HotpotQA, and MuSiQue datasets and conduct the MCTS annotations. We then sample 6,000 correct paths and 6,000 incorrect paths, and the correct paths are used to train the policy model, while both paths are used to train the value model.

Training Process The policy model is initialized using the Qwen2.5-7B-Instruct-1M. Training involves fine-tuning the model for two epochs with a batch size of 1 and a learning rate of 1×10^{-6}, utilizing 8 NVIDIA A100 80GB GPUs. During inference, the expansion size and the beam search size are set to 3.

C Model Sources

- BGE-base-en-v1.5: https://huggingface.co/BAAI/bge-base-en-v1.5
- LLM-Embedder: https://huggingface.co/BAAI/llm-embedder
- ChatGLM3-6B-128K: https://huggingface.co/THUDM/chatglm3-6b-128k
- Qwen2.5-7B-Instruct-1M: https://huggingface.co/Qwen/Qwen2.5-7B-Instruct-1M
- GLM-4-9B-Chat-1M: https://huggingface.co/THUDM/glm-4-9b-chat-1m
- Qwen2.5-14B-Instruct-1M: https://huggingface.co/Qwen/Qwen2.5-14B-Instruct-1M
- Qwen2.5-32B-Instruct: https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
- Qwen2.5-72B-Instruct: https://huggingface.co/Qwen/Qwen2.5-72B-Instruct

D Prompts

Extraction Prompt

```
Extract sentences from the following documents to answer the question. Do not alter the content of the sentences.
Documents: {background}
Question: {query}
Relevant Sentences:
```

Answering Prompt

```
Use only the information from the following documents to answer the question with one short phrase.
Documents: {background}
Question: {query}
Output:
```