# ToT-RAG: Improving Re-ranking of Retrieval-Augmented Generation with Tree of Thoughts

Anonymous EMNLP submission

## Abstract

The Retrieval Augmented Generation (RAG) technique that utilize external knowledge can enable large language models to reduce illusions and perform well in numerous open-domain question answering (ODQA) tasks. The results of re-ranking, as a part of RAG, will be directly used in prompt of the large language model's input, which has a significant impact on the results of RAG system. Therefore, this paper proposes a method of re-ranking based on tree of thoughts (ToT) in RAG, to ensure the overall quality of the text retrieved. This paper not only proposes for the first time to re-rank the texts from multiple dimensions in RAG system, but also combines the large language models with agent to evaluate the text using the tree structure, so that the text obtained from re-ranking would have both outstanding text quality and a high degree of similarity with the user's input. ODQA experiments on three datasets demonstrate that ToT-RAG can effectively reduce illusions and improve the answer accuracy of the RAG system. In comparison experiments, we further illustrate that tree-structured re-ranking is optimal under the trade-off between resource consumption and task accuracy.

## 1 Introduction

The RAG technique(Gao et al., 2023) consists of several important phases including indexing, retrieval, re-ranking, rewriting, and generation. In tasks such as open-domain question and answer, RAG systems enhance Large Language Model (LLM) generation by retrieving relevant documents. However, primitive queries often fail to retrieve the most relevant documents, so improvements to naive RAG methods are needed. For example, DSLR(Hwang et al., 2024) decomposes retrieved documents into individual sentences and filters irrelevant information through sentence-level re-ranking and refactoring. SEER(Zhao et al., 2024) framework trains models to automatically extract evidence information useful for generating tasks through self-aligned learning methods. RaFe(Mao et al., 2024) utilizes existing re-ranking as a feedback mechanism to train query rewriting models to accomplish the task of rewriting a query.The R²AG(Ye et al., 2024) framework bridges the semantic gap between the retriever and the generator using a special former module by introducing retrieval information. Scholars have optimized various aspects of the RAG system and even trained special models to assist in minimizing the illusion of a large language model.

However, few scholars have focused on the optimization of the re-ranking link, whose most general approach is to use the large language models to perform a similarity scoring on the retrieved sentences, which called LLM4Rerank(Gao et al., 2025), and extract the top-ranked sentences to be added to the prompt words answered by the LLM. Nevertheless, this ignores the problem that poor text quality of retrieved sentences or the existence of some irrelevant information reduces the effect of the LLM answer.

Therefore, this paper proposes a multidimensional sentence evaluation system for re-ranking without training any new models. Compared with the mechanism of all dimensions evaluating in parallel, in which there exist results of some dimensions too prominent leading to ignore the results of other evaluation dimensions, or chain-of-rank(Lee et al., 2025) method, our method uses the reasoning ability of the thinking tree structure to organize the various evaluation dimensions, maximize the use of the intelligent ability of agent, and then filter the retrieved sentences on the idea of breadth-first algorithm, to get the best quality of the resorted sentence results. This approach has full potential for ODQA tasks.

Our main contribution can be summarized as following three aspects:

1. We propose a new architecture to do re-

ranking in RAG system, which is called ToT-RAG. It fully utilizes the intelligence of tree structure to do multidimensional evaluation of retrieved texts.

2. We demonstrate that tree-structured multidimensional assessment balances resource utilization and accuracy, which outperforms parallel multidimensional assessment in RAG as a re-ranking module.

3. We show that the best results are the ToT-RAG system with two evaluation dimensions per level of the tree, who outperforms existing benchmarks on the ODQA task on experiment datasets

## 2 Related Work

### 2.1 Advanced RAG

RAG operates through a streamlined process where user queries are first embedded into vector representations and used to retrieve relevant documents from a knowledge base via similarity search. These retrieved documents undergo reranking to prioritize the most pertinent information, which is then processed to fit context windows and integrated into the prompt alongside the original query. RAG has evolved dramatically, with cutting-edge research now focusing on sophisticated retrieval and reranking techniques.

Self-RAG(Asai et al., 2023) proposes to dynamically decide whether to retrieve or not according to the task requirements, and insert special reflection tokens indicating whether external information needs to be retrieved or not as well as self-evaluation of the current generated content during the generation process to get the best retrieval results. CRAG(Yan et al., 2024) training model to evaluate the overall quality of the retrieval results and trigger different knowledge retrieval strategies based on the evaluation results including strategies to supplement knowledge with external web search, semantic chunking of retrieved documents and selective focus on key information to filter out irrelevant content. Moreover, ChunkRAG(Singh et al., 2024) introduces a finer-grained semantic-based block-level filtering mechanism to reduce redundant information interference.

In RAG systems, LLMs are often used as Query Likelihood Models (QLM) to re-rank documents by calculating the probability of generating a query for a given document. However, the direct use of LLMs to approximate QLMs suffers from bias, resulting in estimated distributions that may deviate from the actual document-specific distributions, thus affecting the accuracy of re-ranking.

UR³(Yuan et al., 2024) proposes a novel unsupervised re-ranking framework that improves the reordering performance by maximizing the probability of document generation, unifying the optimization of query generation and document generation under a risk-minimization objective, and evaluating the relevance of each query-document pair independently. RE-RAG(Kim and Lee, 2024) improves the re-ranking quality by introducing an external relevance assessment module "RE" that not only provides relative relevance scores between documents and queries, but also evaluates whether each document actually contributes to answering the query with confidence.

### 2.2 LLM for Ranking

In the field of using LLMs to do ranking, LLMs have emerged as powerful tools for ranking tasks across various domains, with recent innovations addressing previous efficiency and effectiveness challenges. The breakthrough "Pairwise Ranking Prompting" technique introduced by (Qin et al., 2024) has demonstrated that even moderate-sized open-source LLMs can achieve state-of-the-art ranking performance, outperforming larger commercial models by focusing on relative comparisons between pairs rather than absolute scoring of items.

Meanwhile, research on multi-conditional ranking has advanced through the MCRank benchmark (Pezeshkpour and Hruschka, 2025), which evaluates LLMs' capabilities in handling complex ranking scenarios with multiple, sometimes conflicting criteria. This benchmark has revealed that while LLMs struggle with increasing complexity of items and conditions, novel decomposed reasoning methods like EXSIR can significantly enhance performance.

These developments mark substantial progress in adapting LLMs for practical ranking applications in retrieval systems, recommendation engines, and information organization tasks, where efficiency and multi-criteria decision-making are crucial considerations.

### 2.3 Tree of Thoughts

ToT structure represents a groundbreaking advancement in reasoning methodologies that enhances

LLMs' problem-solving capabilities by structuring reasoning as an explorable decision tree rather than a linear sequence. Unlike CoT(Wei et al., 2022), which follows a single reasoning path, ToT creates multiple intermediate reasoning branches which are called "thought", evaluates their promise, and strategically explores the most viable paths while pruning unpromising ones—mirroring human deliberative thinking.

These advancements collectively demonstrate ToT's versatility across diverse applications including complex problem-solving, multi-conditional ranking and so on.

## 3 Method

### 3.1 Overview of Tree of Thought

The Tree of Thoughts framework(Yao et al., 2023) is designed to enhance LLMs' problem-solving abilities by structuring their reasoning in a manner similar to human cognitive processes, and consists of four key components.

First, idea decomposition is the explicit breaking down of a problem into smaller steps called ideas. Second, for the input $x$ and the thought state $s = [x, z_{1\ldots i}]$, there are two main techniques for the generation of thoughts $G(p_\theta, s, k)$, one is independent sampling $z^{(j)} \sim p_\theta^{CoT}(z_{i+1} \mid s) = p_\theta^{CoT}(z_{i+1} \mid x, z_{1\ldots i})(j = 1\ldots k)$, the second is sequential generation of $[z^{(1)} \ldots z^{(k)}] \sim p_\theta^{propose}(z_{i+1}^{(1\ldots k)} \mid s)$. Again, the state is evaluated $V(p_\theta, S)$, two strategies are commonly used for this purpose, one is value based $V(p_\theta, S)(s) \sim p_\theta^{value}(v \mid s) \forall s \in S$, the second is voting $V(p_\theta, S)(s) = \mathbf{1}[s = s^*]$. Finally, the search algorithm obtains the optimal path to solve the task, based on the tree structure two basic algorithms are usually used, they're Breadth-First Search (BFS), and Depth-First Search DFS (DFS). Therefore, the tree completes an inference task.

### 3.2 ToT-RAG framework

The idea of our re-ranking optimization method in RAG can be described in the following form:

Let the set of candidate sentences be $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, and let the user query be denoted as $q$.

We define a set of base evaluation functions as $\{f_1, f_2, \ldots, f_k\}$, where each $f_i(c_j, q)$ returns a score for sentence $c_j$ under quality dimension $i$.

The tree of thoughts $\mathcal{T}$ is a hierarchical composition function:

$$\mathcal{T}(c_j, q) = Q(c_j) = F(f_1(c_j, q), \ldots, f_k(c_j, q))$$

where $F$ is an aggregation tree-structure function.

We define the final set of high-quality selected sentences as:

$$\mathcal{C}^* = \{c_j \in \mathcal{C} \mid Q(c_j) \geq \tau\}$$

where threshold $\tau$ generated from assessment models.

In general, the objective is to maximize the overall quality of the selected set:

$$\max_{\mathcal{C}^*} \sum_{c \in \mathcal{C}^*} Q(c)$$

As shown in Figure1, given the sentences retrieved from the RAG system, we first divide the re-ranking assessment into two steps, where the text quality is assessed in terms of fluency, accuracy, completeness, conciseness, and novelty at the first level of the thought tree. After the evaluation and screening of the first layer of thoughts, the remaining sentences are subjected to the second layer of text evaluation. This can be formulated as:

Given tools $t_i \in \Theta \quad (i = 1, 2, 3, 4, 5)$, which are used to generate thoughts. Agents sample thoughts $z$ as $z^{(t_1, t_2)} \sim p_\Theta$ to form new state $s$. Then thought evaluators $V(p_\Theta, S)$ are used to filter sentences.

The second layer of the thought tree mainly evaluates the retrieved sentences and user inputs for similarity in terms of lexics, semantics, pragmatics, structure, style, etc., then evaluates and filters the second layer of the thought which is similar to the former layer, and the final sentences remained are injected into the prompt words of the large language model to assist the downstream tasks by reducing the hallucination.

### 3.3 Implementation

**Information Flow**

In a typical RAG pipeline, the process begins with query processing, where user inputs are transformed into vector representations using dense encoders such as SBERT or E5 embeddings for effective semantic matching. The retrieval phase then leverages hybrid approaches combining sparse methods like BM25 (Askari et al., 2023) with dense vector similarity search to identify relevant documents from knowledge bases.

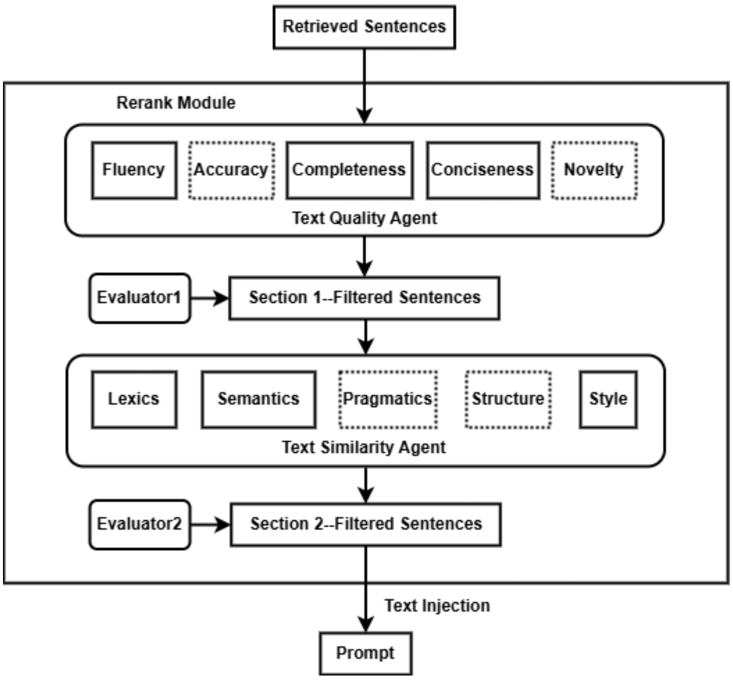

Figure 1: ToT-RAG Framework

Retrieved documents undergo contextual integration through relevance-based reranking with cross-encoders, while iterative refinement approaches like those in Layer-of-Thoughts (Fungwacharakorn et al., 2024) progressively filter and structure information through hierarchical reasoning layers to produce more accurate, comprehensive, and trustworthy outputs, and this is also where we are working on optimizing.

### Models Preparation

Given retrieval texts, models that evaluate texts in each dimension separately are the basis of our methodology. When generating thoughts, the models are all LLMs, but they score the utterances in each of the above dimensions by prompt engineering, limiting the scores to between 0 and 1.

When evaluating each layer of ideas in the tree, we similarly use the LLM to evaluate the results produced by themselves in conjunction with its own capabilities, and set up thresholds for scoring in each evaluation dimension. Statements above the thresholds indicate that they passes the test in corresponding dimensions and will be retained, while statements below the thresholds will be filtered out.

### Tool Selection

As illustrated by researchers(Ferrag et al., 2025), LLM agents, leveraging LLMs as their cognitive core, represent an emerging paradigm that transforms passive text generators into autonomous systems capable of planning, decision-making, and tool manipulation to achieve complex goals.

We encapsulate the models evaluating text quality and text similarity in different agents, and each agent plays a role in different stages of the thought tree, maximizing their intelligence to select and call from the encapsulated models for evaluating each dimension.

The call is based on the fact that if each sentence performs well in a certain dimension, the priority of that dimension will be lowered, and if the performance of each sentence varies greatly in a certain dimension, the priority of that dimension will be high, so as to ensure that the final filtered sentence performs well in all dimensions.

### Prompt Injection

In RAG system, query rewriting combined with reranked texts represents a crucial enhancement to the traditional RAG pipeline. This process transforms the way language models interact with external knowledge.

Ultimately, the final goal is to determine the optimal re-ranking text result according to the idea of breadth-first search (BFS). The idea implicit in our method is to traverse each layer of nodes which equal to the evaluation dimensions in the tree with a node limit set to be two, filter the sentences through the first layer of the tree for text quality assessment, and the remaining sentences go to the second layer of the tree for similarity assessment, and the retained sentences are rewritten into the prompt and inputted into the LLM to be used as task generation.

## 4 Experiments

The system we designed is based on a two-layer tree structure, with each layer judging the text based on two dimensions of agent calls, and the sub-models we use are mainly gpt-4o-mini and gpt-4.1-nano. We test the performance of our ToT-RAG system in an open-domain question and answering task using three datasets and compare the results with benchmark, finding that our approach is effective in improving answering accuracy.

### 4.1 Datasets

Datasets used are PubHealth(Zhang et al., 2023) (true-or-false question), PopQA(Mallen et al., 2023) (long-tail short-form answer) and TriviaQA(Joshi et al., 2017) (common short-form answer), whose question-answer pairs are all reviewed and annotated. Each question is followed by a standardized answer and at least twenty related texts.

**PubHealth**

The dataset is a comprehensive resource for public health misinformation verification, containing real-world health-related claims collected from fact-checking websites like Snopes and Politifact. Each claim is meticulously labeled with veracity class and accompanied by journalist-crafted explanations that justify the fact-check assessment.

**PopQA**

The dataset deliberately include of "long-tail" knowledge—less common facts that may not be well-represented in model training data. It was constructed through a weighted sampling of knowledge triples from the C4 corpus, ensuring a balance between popular entities and more obscure information, making it an effective benchmark for assessing LLMs' factual reliability.

**TriviaQA**

The dataset features naturally occurring trivia questions authored by enthusiasts, each paired with several independently gathered supporting documents from Wikipedia and the web. What distinguishes TriviaQA is its organic question creation process, completely decoupled from the evidence collection, which helps minimize potential biases while ensuring genuine question complexity.

### 4.2 Metrics

In order to evaluate the performance of our proposed RAG system in reducing modeling illusions and improving question-answer accuracy in the ODQA task. Following previous work(Kim and Lee, 2024; Yan et al., 2024; Singh et al., 2024), accuracy was adopted as the evaluation metric for all datasets. Which can be formulated as:

$$\text{Accuracy} = \frac{\text{Number of Correct Generations}}{\text{Total Number of Generations}} \tag{1}$$

### 4.3 Results Analysis

As shown in Table1, we compare the generation abilities of our ToT-RAG system with outstanding LLMs or other retrieval question-answer RAG systems, yielding the following brief insights:

**General Enhancement**

Our model has substantially improved performance over retrieve-free LLMs (e.g., LLaMA, Alpaca, GPT) and significantly outperforms the standard RAG with LLMs approach.

So ToT-RAG greatly compensates for the lack of factual knowledge and contextual support in pure language models, verifying the importance of retrieval enhancement and inference mechanisms for complex QA tasks. Based on the standard RAG framework, the depth of reasoning and the accuracy of evidence selection are further enhanced, which is an important evolution of traditional RAG methods.

**Performance Beyond Advanced RAG**

ToT-RAG system shows strong and stable performance in several tasks with certain generalization ability, especially in complex tasks that require multi-hop reasoning or information fusion. This suggests that the introduction of a re-ranking strategy based on the tree-based reasoning mechanism can help integrate retrieved evidence more efficiently and improve the accuracy and reliability of generated answers.

On the TriviaQA dataset, ToT-RAG achieves an accuracy of 78.7, which significantly outperforms all baseline methods. The dataset is known for its multi-hop inference and difficult factual integration, and ToT-RAG's performance proves its superiority in complex inference scenarios. On the PopQA

Table 1: Performance Comparison on PubHealth, PopQA and TriviaQA

| **Method** | **PubHealth** | **PopQA** | **TriviaQA** |
|---|---|---|---|
| **(A) LLMs Without Retrieval** | | | |
| LLaMA2-7B | 34.2 | 14.7 | - |
| Alpaca-7B | 49.8 | 23.6 | - |
| LLaMA2-13B | 29.4 | 14.7 | - |
| Alpaca-13B | 55.5 | 24.4 | - |
| ChatGPT | 70.1 | 29.3 | - |
| **(B) Standard RAG with LLMs** | | | |
| RAG + LLaMA2-7B | 30.0 | 38.2 | - |
| RAG + Alpaca-7B | 40.2 | 46.7 | - |
| RAG + LLaMA2-13B | 30.2 | 45.7 | - |
| RAG + Alpaca-13B | 51.1 | 46.1 | - |
| **(C) Advanced RAG** | | | |
| RAG | 39.0 | 52.8 | - |
| Self-RAG | 72.4 | 54.9 | 66.4 |
| CRAG | 75.6 | 59.8 | - |
| Self-CRAG | 74.8 | 61.8 | - |
| ChunkRAG | **77.3** | 64.9 | - |
| ChatGPT+RE | - | - | 77.7 |
| **ToT-RAG** | 73.2 | **66.1** | **78.7** |

dataset, ToT-RAG also performs well, outperforming the currently strongest ChunkRAG (64.9) and Self-CRAG (61.8) with an accuracy of 66.1, indicating that ToT-RAG's reasoning and information selection capabilities are more adaptable to open-domain question and answering scenarios. On the PubHealth dataset, ToT-RAG scores 73.2, which is slightly lower than ChunkRAG (77.3) and CRAG (75.6), but still far exceeds the standard RAG and LLM baseline. Considering that this dataset requires high precision for medical facts, ToT-RAG's performance is still at an advanced level, indicating that its reranking strategy also has some advantages in the professional field.

It is thus revealed that the re-ranking idea proposed in this paper makes the retrieved text more robust and makes the text generation task more effective especially in short-form answer generation.

## 5 Comparison Study

In order to verify the validity of the tree structure proposed in this paper for generating assessment thoughts and the association between the tree structure and the QA generation task, the following exploratory experiments are set up in this paper.

### 5.1 The Need for a Tree Structure

The general ranking task evaluates the text in one dimension, but it is conceivable that weak performance in other dimensions of the text would make it difficult to analyze it, and thus injecting it with prompt would affect the generation of a LLM. It is easy to think of extending the evaluation dimensions to multiple dimensions, but in contrast to ordinary parallel evaluation of multiple dimensions, we have proposed to organize the generation of evaluation thoughts in the structure of a tree, and the following experiments are designed on the PopQA dataset to verify the necessity of this model design.

We let agent automatically call two non-repeating analysis tools following our ToT-RAG system as the experimental group, and set the evaluation of all the models called in sequence as the control group, record the number of tokens and response time used in each response while recording the results of each pair of QA tasks, and finally calculate the average token consumption and time consumption, the experimental results are shown in Figure2 and Figure3.

**Analysis**

From the figures, it can be seen that the accuracy of the task based on agent invoking the model is

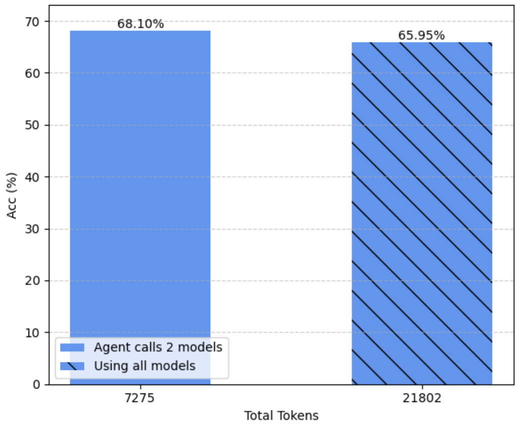

Figure 2: Token Comparison Result

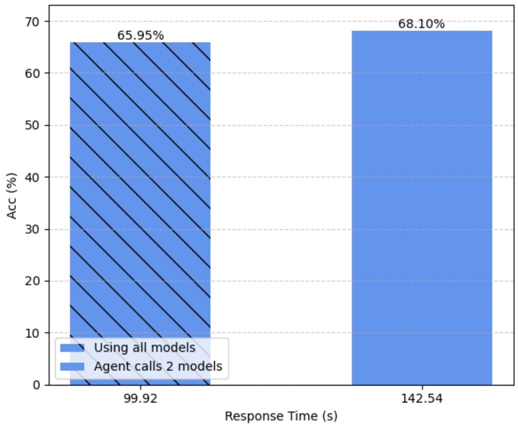

Figure 3: Time Comparison Result

2.15% higher than directly using all models, and the number of tokens it uses is about one-third of the number of invoking all models, although the response time is about 40% higher, which can be interpreted as a result of the fact that in order to achieve the best model results, the agent is maximizing the use of its own intelligence to analyze the problem of calling the model, and therefore lengthening the response time.

Although the time consumption required for agent to invoke the model is greater, the space consumption required to invoke all models is orders of magnitude greater than that required for agent to invoke the model. And ultimately the tree-structure text analysis model of the agent calling can achieve a higher accuracy rate, it can be considered that this method proposed in our paper in the use of time and space resources to achieve a balance, which is helpful to improve the model response effect.

## 5.2 Explore Optimal Number of Nodes

Given that the structure of the tree does help the reranking session of the RAG system, the structure of the tree will be further explored in this section. the agent is considered based on the necessity of calling each model, and each level of the tree is used by the agent to call the text evaluation models according to the order of priority, so how many evaluation models can be used to make the reordered text play the optimal utility? We designed the following comparative experiments which are also designed in the PopQA dataset.

The base number of evaluation models which also means the number of nodes in each layer of our thinking tree is five. So we design experiments using the agent to call one to four evaluation models in each layer of the tree for cross-checking experiments. We want to explore the optimal number of invoked nodes according to the recorded results of task accuracy and time consumption following the change of the number of nodes, which can be seen in Figure4 and Figure5.

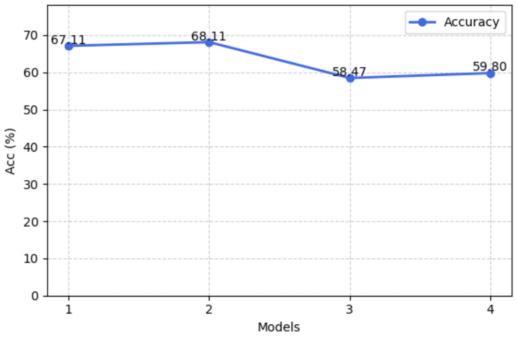

Figure 4: Accuracy Trend over Nodes

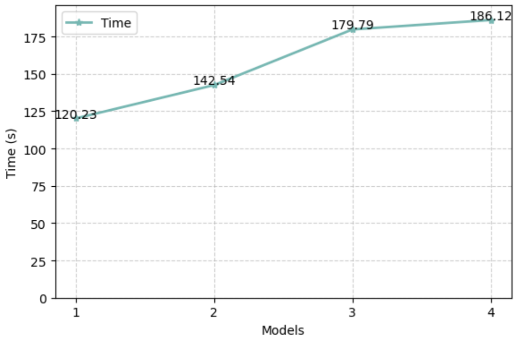

Figure 5: Time Consumption Trend over Nodes

### Analysis

From the figure, it can be seen that at each layer of the tree as the number of agents called models increases from one to the other, the model accuracy rate experiences an increasing and then decreasing trend, and when the number of nodes in the tree is two, which means that when two evaluation models are called at each layer, the accuracy of the model is the highest and reaches 68.11%. And when the number of models increases further, the effect of the model decreases significantly, even less than 60%. And the average time consumption of each quiz is gradually increasing with the number of thoughts in the tree, compared with the least time consuming which only calls one model, agent call two models thinking can improve the task accuracy within the acceptable range of time consumption.

So we find that when the base number of models per layer of the tree is five, the agent calling two models for the analysis is the best.

## 6 Conclusion

In this paper, as a re-ranking optimization of the RAG technique, we propose a tree-of-thoughts based re-ranking technique, which has the advantage of making full use of the reasoning ability of the tree structure for multi-dimensional evaluation of sentences, so that the final statements obtained by retrieving and re-ranking from external knowledge have the advantage of all the dimensions, which serves as the complementary knowledge to help LLMs to understand the unfamiliar knowledge, to reduce the illusions and to enhance the LLMs' generation capability.

Experiments show that the method proposed in this paper has some generalization ability on the ODQA tasks, and the task accuracy is able to exceed the strongest existing RAG method which also improves retrieval results to reach the 1st place on multiple datasets. Meanwhile, validation experiments show that the tree structure does work in a multidimensional ranking system, as well as the best text generation results can be achieved when the tree nodes are taken two.

## Limitations

Although the ToT-RAG model proposed in this paper enhances the re-ranking ability of the RAG system, the time required to generate the results is long so the agent calling session needs to be optimized. In addition, the selection of the tree structure in this paper relies on experimental decisions and lacks universality on a wider range of tasks, looking forward to future adaptive research on tree structure based on this study, which will help to generalize the thinking-tree based ranking method. Finally the implementation of the model in this paper relies on the LLM, the stability of the LLM may affect the textual research results, and different application environments may also lead to different results. However, despite the above shortcomings, the method proposed in this paper is significant in improving the quality of texts obtained from re-ranking in a prospective and comprehensive way.

## Ethics Statement

There may be potential ethical issues with the answers generated by the LLMs, but the LLM quizzes involved in the experiments in this paper are retrieval-augmented generation within the scope of publicly available datasets and do not involve any harmful segments. Moreover, the datasets used in this paper are publicly available benchmark datasets and there is no conflict of interest with any individual or organization.

## References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the bm25 score as text improves bert-based re-rankers. In *European Conference on Information Retrieval*.

Mohamed Amine Ferrag, Norbert Tihanyi, and Mérouane Debbah. 2025. From llm reasoning to autonomous ai agents: A comprehensive review.

Wachara Fungwacharakorn, Ha-Thanh Nguyen, May Myo Zin, and Ken Satoh. 2024. Layer-of-thoughts prompting (lot): Leveraging llm-based retrieval with constraint hierarchies. *ArXiv*, abs/2410.12153.

Jingtong Gao, Bo Chen, Xiangyu Zhao, Weiwen Liu, Xiangyang Li, Yichao Wang, Wanyu Wang, Huifeng Guo, and Ruiming Tang. 2025. LLM4rerank: LLM-based auto-reranking framework for recommendations. In *THE WEB CONFERENCE 2025*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *ArXiv*, abs/2312.10997.

Taeho Hwang, Soyeong Jeong, Sukmin Cho, SeungYoon Han, and Jong Park. 2024. DSLR: Document refinement with sentence-level re-ranking and reconstruction to enhance retrieval-augmented generation. In *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 73–92, Bangkok, Thailand. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *ArXiv*, abs/1705.03551.

Kiseung Kim and Jay-Yoon Lee. 2024. RE-RAG: Improving open-domain QA performance and interpretability with relevance estimator in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22149–22161, Miami, Florida, USA. Association for Computational Linguistics.

Juntae Lee, Jihwan Bang, Kyuhong Shim, Seunghan Yang, and Simyung Chang. 2025. Chain-of-rank: Enhancing large language models for domain-specific RAG in edge device. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5601–5608, Albuquerque, New Mexico. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2024. RaFe: Ranking feedback improves query rewriting for RAG. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 884–901, Miami, Florida, USA. Association for Computational Linguistics.

Pouya Pezeshkpour and Estevam Hruschka. 2025. Multi-conditional ranking with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2863–2883, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Ishneet Sukhvinder Singh, Ritvik Aggarwal, Ibrahim Allahverdiyev, Muhammad Taha, Aslihan Akalin, Kevin Zhu, and Sean O'Brien. 2024. Chunkrag: Novel llm-chunk filtering method for rag systems. *arXiv preprint arXiv:2410.19572*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601.

Fuda Ye, Shuangyin Li, Yongqi Zhang, and Lei Chen. 2024. $\mathrm{R}^2$AG: Incorporating retrieval information into retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11584–11596, Miami, Florida, USA. Association for Computational Linguistics.

Xiaowei Yuan, Zhao Yang, Yequan Wang, Jun Zhao, and Kang Liu. 2024. Improving zero-shot LLM reranker with risk minimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17967–17983, Miami, Florida, USA. Association for Computational Linguistics.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen M. Meng, and James R. Glass. 2023. Interpretable unified language checking. *ArXiv*, abs/2304.03728.

Xinping Zhao, Dongfang Li, Yan Zhong, Boren Hu, Yibin Chen, Baotian Hu, and Min Zhang. 2024. SEER: Self-aligned evidence extraction for retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3027–3041, Miami, Florida, USA. Association for Computational Linguistics.

# A Prompt Templates

### Prompt for quality assessment

In Each Dimension:

As a language expert, please rate the {dimension} of the following sentence (the range is a decimal between 0 and 1, where 1 indicates complete {dimension} and 0 indicates very incoherent). Sentence: {sentence} Please only return a decimal fraction and do not attach any explanations.

### Prompt for similarity assessment

In Each Dimension:

As a language expert, please evaluate the {dimension} between each input sentence and the user's input query respectively (the range is a decimal between 0 and 1, where 1 indicates completely {dimension} and 0 indicates not similar). {dimension} means :. input sentence: {sentence} user's input query: {query} Please only return a decimal fraction and do not attach any explanations.

### Prompt for thought evaluator

You are a global evaluation large language model used to assess the local multi-dimensional scoring results of gpt-4.1-nano. Please use the knowledge you have to set thresholds for the scoring results of each dimension respectively. Sentences with scores higher than the thresholds should be retained to indicate high text quality, while those with scores lower than the thresholds should be discarded to indicate low text quality. Also, it is hoped that the number of sentences retained at the end will be as much as half or more of the original sentences. I also hope that as many sentences as possible will be retained at the intersection after the screening of the three models. {actions} are several evaluation dimensions. You need to set thresholds for each of them respectively. {observations} are the scoring results corresponding to each dimension of each sentence for your reference. The range of each threshold is a decimal between 0 and 1. Please only return three decimal fraction and do not attach any explanations. And output it in the form separated by English commas.

### System message for final LLM chat

You are a helpful assistant that is an expert at extracting the most useful information from a given text. Also bring in extra relevant information to the user query from outside the given context. If you're confused about the user's query, you'd better answer based on the konwledge of given context instead of hallucinating. If the user just wishes to greeting with you, introduce yourself is enough!