Tree-of-Question: Structured Retrieval Framework for Korean Question Answering Systems

Dongyub Lee[1]*, **Younghun Jeong**[1]*, **Hwayeon Kim**[1]*, **Hongyeon Yu**[1]*, **Seunghyun Han**[1],
Taesun Whang[1], **Seungwoo Cho**[1], **Chanhee Lee**[1], **Gunsu Lee**[1], **Youngbum Kim**[1]†
[1] Naver Corp, WA, USA
{dongyub.lee,younghun.j,hwayeon.kim,hongyeon.yu,youngbum.kim}@navercorp.com

Abstract

We introduce Korean language-specific RAG-based QA systems, primarily through the innovative Tree-of-Question (ToQ) methodology and enhanced query generation techniques. We address the complex, multi-hop nature of real-world questions by effectively integrating advanced LLMs with nuanced query planning. Our comprehensive evaluations, including a newly created Korean multi-hop QA dataset, demonstrate our method's ability to elevate response validity and accuracy, especially in deeper levels of reasoning. This paper not only showcases significant progress in handling the intricacies of Korean linguistic structures but also sets a new standard in the development of context-aware and linguistically sophisticated QA systems.

1 Introduction

Recent advancements in Large Language Models (LLMs) have revolutionized information seeking and text generation, with real-world applications such as Bing Chat[1], Perplexity.ai[2], and Google Bard[3] leading the way. These systems utilize the Retrieval Augmented Generation (RAG) methodology, where responses are crafted based on information extracted from retrieved documents. This approach allows these platforms to provide answers that not only are contextually relevant but also cite the sources they reference, enhancing the reliability and transparency of the information provided. This strategy is particularly effective in addressing the inherent issue of hallucinations in LLMs, where LLMs might generate plausible but factually incorrect information.

Meanwhile, a system named "Naver Cue:"[4] (Yu et al., 2023) has emerged as a noteworthy addition to the realm of question and answering systems, specifically tailored to the Korean language. It is a system grounded in HyperCLOVAX[5] (Kim et al., 2021; Shin et al., 2022; Yoo et al., 2024), a Korean Large Language Model with billions of parameters. Utilizing the RAG methodology and leveraging Naver's search engine, "Naver Cue:" efficiently provides contextually relevant and accurate responses. The integration of HyperCLOVAX into this system enables a sophisticated understanding and processing of Korean language nuances, ensuring precise and relevant answers. This makes "Naver Cue:" a notable advancement in the domain of LLM-based QA systems tailored for Korean language.

In the realm of RAG-based QA systems powered by LLMs, the operational mechanism typically unfolds in a structured three-step process. Initially, the system generates a query derived from the user's question, effectively translating the user's inquiry into a format suitable for search engines. Following this, the query is processed through the search engine, which conducts a comprehensive search to gather relevant information. The final step involves generating a response by synthesizing and summarizing the search results into a coherent answer.

A critical aspect of this process is the creation of queries that are specifically optimized for each search engine. This optimization is crucial as it directly influences the relevance and accuracy of the information retrieved. Furthermore, the nature of questions posed by users in such systems is predominantly multi-hop. These multi-hop questions are multifaceted, often characterized by their inherent ambiguities or the need to collate information from multiple sources (Mavi et al., 2022; Amplayo et al., 2022; Trivedi et al., 2022). This complexity poses additional challenges in query formulation, making it imperative for the RAG-based QA systems

* Equal contribution
† Corresponding Author
[1] https://www.bing.com/chat
[2] https://www.perplexity.ai
[3] https://bard.google.com
[4] https://cue.search.naver.com
[5] https://clova.ai/hyperclova

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 406–418
June 16-21, 2024 ©2024 Association for Computational Linguistics

to have advanced understanding and processing capabilities.

Particularly in the case of the Korean language, accurately interpreting the user's question becomes more challenging due to the duality and connotation of words (Park et al., 2020; Kim et al., 2021). Korean often involves subtle nuances and implied meanings that can significantly alter the context of a query. Additionally, Korean is an agglutinative language, characterized by its unique grammatical structure where particles follow nouns, and the stems of verbs or adjectives are followed by endings. These endings express various grammatical properties, adding layers of complexity to the language (Lee et al., 2020; Yang, 2021; Son et al., 2022). These linguistic features can lead to difficulties in generating search queries that precisely mirror the user's intent. Therefore, RAG-based QA systems designed for Korean must possess enhanced capabilities to discern and reflect these subtleties.

Recent research in closed-book QA systems utilizes metrics like ROUGE-L and Disambig-F1 for performance evaluation, comparing model predictions with ground truth answers (Lin, 2004; Amplayo et al., 2022). These metrics assess end-to-end performance, highlighting how closely model responses match expected answers. However, they fall short in evaluating specific aspects crucial to LLM-based QA systems such as query generation, document retrieval, and response generation. In real-world scenarios, where correct answers aren't predetermined, this becomes a challenge. To address these limitations, new metrics like citation recall/precision (Gao et al., 2023) and FactScore (Min et al., 2023) have been introduced. These focus on evaluating the system's ability to reference relevant documents and the relevance of summarized responses to posed questions.

Despite these advancements, a notable gap remains: there is currently no established metric for evaluating the appropriateness of the queries generated by the in-house search engine in response to the user's questions. This highlights a crucial area for further research and development, as the ability to generate accurate and relevant queries is fundamental to the success of real-world QA systems.

To effectively address the challenges in the current landscape of LLM-based Open-domain QA systems, particularly for the Korean language, our research introduces several pivotal contributions, summarized as follows:

- **Enhanced Query Generator in Korean RAG-based Long-form QA Systems:** We propose an advanced role for the Query Planner, optimizing queries from user inquiries for search engine compatibility. This aims to improve the accuracy and relevance for Korean language nuances.

- **Tree-of-Question for Multi-Hop Reasoning:** Introducing a structured Tree-of-Question concept, our approach enhances the system's capacity to process multi-hop questions in Korean.

- **Novel Evaluation Method for Query Planner:** We develop a new method for evaluating the Query Planner in multi-hop query processing systems, utilizing LLMs for both offline and online assessments.

2 Task Definition

In LLM-based Retrieval Augmented Generation (RAG) QA systems, we define the process as a sequence of functions, each transforming an input to produce an output that serves as the input for the subsequent function. Let q be the user's question. Then, the process can be formalized using the following notation:

1. *Query Generation*: A function f_{QG} takes q and generates a query Q.

$$Q = f_{QG}(q)$$

This involves interpreting q and restructuring it into a format optimized for the in-house search engine.

2. *Document Retrieval*: A function f_{DR} takes Q and retrieves a set of documents D from the in-house search engine.

$$D = f_{DR}(Q)$$

These documents are relevant to the query and contain information pertinent to answering q.

3. *Response Generation*: A function f_{RG} takes q, Q, and D, and generates a final response R.

$$R = f_{RG}(q, Q, D)$$

This involves synthesizing information from D in the context of q and Q to provide a comprehensive and accurate answer to the user's question.

In our research, we emphasize the development of an effective *Query Generator* within the LLM-based RAG QA framework. We operate under the assumption that the in-house search engine and the *Response Generator* are already established and functional. Our focus is primarily on enhancing the *Query Generator*, f_{QG}, which is responsible for transforming user questions q into optimized queries Q.

3 Enhanced Query Planning with Tree-of-Question and Query Evaluator

In the Korean RAG-based long-form QA system, complex questions often require structuring into simpler, searchable queries. Previous studies have explored various aspects of multi-hop reasoning in closed-book QA systems. In the study by (Min et al., 2019), question types requiring multi-hop reasoning were classified into three primary categories: bridging, intersection, and comparison. Furthermore, the research conducted by (Amplayo et al., 2022), which draws inspiration from reasoning chains in LLMs. This approach entails formulating explanations as a sequence of interconnects. Additionally, Trivedi et al. (2022); Press et al. (2022) proposed methods of generating questions sequentially and creating follow-up questions when necessary. This sequential approach is valuable for developing a deeper understanding of the topic in question and ensuring comprehensive coverage.

However, these methods encounter limitations in responding to questions that search for multiple queries in parallel and then synthesize these answers. Such complex scenarios, referred to as the "Hybrid" type in Table 1, require a more nuanced approach that combines elements of different reasoning types. This gap highlights the need for advanced methodologies capable of handling these hybrid multi-hop reasoning challenges effectively.

3.1 Tree of Questions

As illustrated in Figure 1, we propose a novel method, Tree-of-Question (ToQ), to decompose and structure complex queries in a tree-like format. This approach is inspired by the multi-hop question concept in QA and the Tree-of-Thought (ToT) methodology from recent advancements in LLMs (Yao et al., 2023). Our system logically connects and structures questions to facilitate planned retrieval and comprehensive search processes in multi-hop reasoning scenarios.

Algorithm 1 illustrates the process of the ToQ method. The Root node represents the user's original question. Each node in the tree corresponds to a sub-question derived from or related to the original query or its preceding nodes. The level of a node indicates its depth in the tree, representing the sequential steps needed to reach a conclusive answer. The ToQ dynamically expands as it decomposes complex questions into simpler, interconnected sub-questions. When a dependency between Nodes is identified, especially in cases similar to Bridging or Hybrid scenarios, as illustrated in Table 1, the Answer Integrator is utilized to fill in the necessary answers in the [ANS] portion of the question. Finally, the ToQ process terminates when the original user's question can be satisfactorily answered using the responses from the created nodes, a determination made using the Query Evaluator.

Algorithm 1 Tree of Questions

```
$Root$: The original question ($Q$)
$Level$: The depth in the tree, representing the number of nodes to reach an answer

function TreeOfQuestion(Node)
    Determine $Level$ of $Node$ and increment hop count
    Identify dependent $Node$s, use Answer Integrator if related to parent $Node$ and modify $Root$ or $Node$ if needed
    Generate query from the (modified) $Root$ or $Node$
    Generate response based on the query and retrieved documents
    $Eval$ ← Evaluate query and response using Query Evaluator
    if $Eval$ is positive then
        return
    end if
    $DecomposedNodes$ ← Decompose $Node$ into sub-questions if needed
    for each $SubNode$ in $DecomposedNodes$ do
        Create new $Node$ for $SubNode$
        Update $Level$ for new $Node$
        Recursively call TreeOfQuestion($SubNode$)
    end for
end function
```

3.2 Answer Integrator

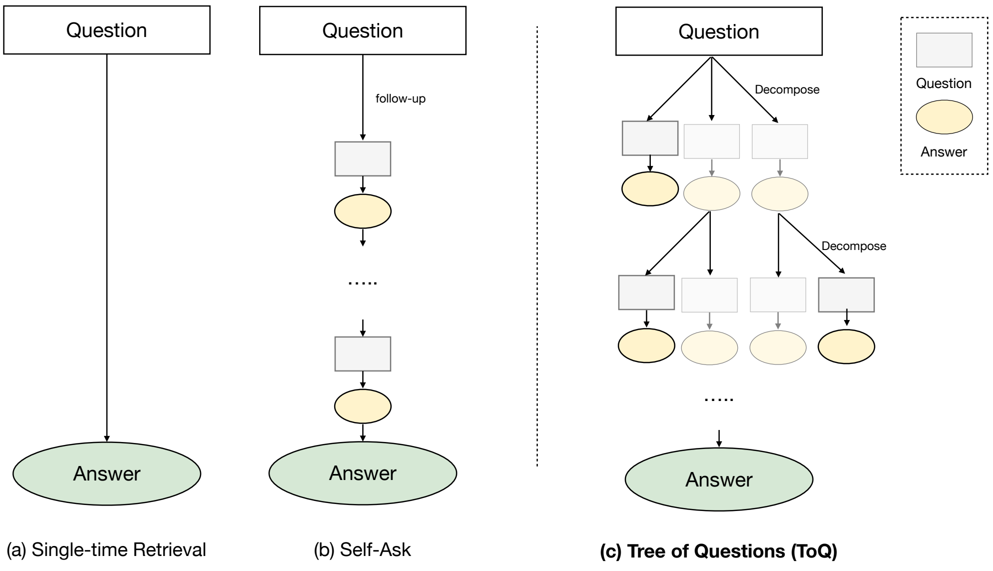
The answer integrator is designed to precisely identify and extract the answer span from a document that aligns with the original query's intent. It functions by analyzing the relevance between a user's question and the provided document. If a relevant match is found, the Answer Integrator extracts the specific answer span from the document. The instruction prompt of Answer Integrator is described in Appendix Prompt B.

3.3 Query Generator

Utilizing a few-shot example-based approach, the query generator model adeptly transforms user

Type	Details
Bridging	**Complex Question (Korean)**: BMW i5와 비슷한 가격대의 전기차 추천해주세요. **Translation**: Recommend an electric car in a similar price range to the BMW i5. **Structured Questions**: Q1: BMW i5 price range → 120 million Q2: Electric vehicles in [Q1_ANS] price range.
Intersection	**Complex Question (Korean)**: 놀란 감독의 작품 중 오펜하이머가 출연한 영화가 있나요? **Translation**: Are there any films by Director Nolan starring Oppenheimer? **Structured Questions**: Q1: Oppenheimer's filmography Q2: Director Nolan's filmography
Comparison	**Complex Question (Korean)**: 갤럭시랑 아이폰 중 어느 핸드폰이 배터리 수명이 더 긴가요? **Translation**: Which phone has a longer battery life, Galaxy or iPhone? **Structured Questions**: Q1: Galaxy's battery life Q2: iPhone's battery life
Hybrid	**Complex Question (Korean)**: 캐리비안의 해적 시리즈중 제일 관객수가 많은게 뭐야? **Translation**: Which of the Pirates of the Caribbean series has the largest audience? **Structured Questions**: *Bridging* Q1: Pirates of the Caribbean series → Pirates of the Caribbean 1, 2, 3, 4, 5. Q2: Series with the largest audience among [Q1_ANS] *Comparison* Q3: Pirates of the Caribbean 1 audience numbers → 656.3 million Q4: Pirates of the Caribbean 2 audience numbers → 1.044 billion Q5: Pirates of the Caribbean 3 audience numbers → 960 million Q6: Pirates of the Caribbean 4 audience numbers → 865 million Q7: Pirates of the Caribbean 5 audience numbers → 794 million

Table 1: Examples of Multi-Reasoning Type Questions in Korean.



(a) Single-time Retrieval

(b) Self-Ask

(c) Tree of Questions (ToQ)

Figure 1: Architecture of Tree of Questions.

questions into optimized queries for retrieving relevant documents from an in-house search engine. Given k in-context exemplars of question-query pairs $[(q_1, Q_1), \ldots, (q_k, Q_k)]$, along with an instruction, the query generator generates a query Q for a question q. This process is pivotal in accurately sourcing information, ensuring that the generated queries Q are precisely formulated to align with the user's inquiry q, as demonstrated in Appendix Prompt B.

3.4 Query Evaluator

The query evaluator, as utilized in line 9 of Algorithm 1, plays a crucial role in determining if the original question, denoted as q, is adequately addressed by the generated queries Q and responses. It uses a LLM to evaluate these elements on four key aspects. **Semantic Coherence** assesses the logical flow and relevance of the response to q, scored from 1 (no coherence) to 10 (perfect coherence). **Answerability** measures the likelihood of the response directly addressing q, with a confidence level expressed as a percentage from 0% to 100%.

Each response's **Overall Assessment Score** is computed by averaging the Semantic Coherence score and the Answerability score (after converting it from a percentage to a 1-10 scale). The evaluator also provides a **Response Validity** indicator, a binary (true/false) metric that determines the adequacy of responses in answering q based on coherence and answerability assessments. This indicator is crucial in determining the applicability of the responses to real-world questions, providing a clear binary decision. Details about the evaluator's instruction prompt can be found in Appendix Prompt B.

4 Experiments

4.1 Dataset Collection

Existing English datasets for multi-hop QA, such as HotpotQA (Yang et al., 2018), 2WikiMultihopQA (Ho et al., 2020), and ASQA (Stelmakh et al., 2022), provide a foundation for evaluating multi-hop QA on English benchmarks. These datasets consist of questions and their corresponding answers in a closed-book setting, focusing on generating accurate answers to given questions and documents.

To address the absence of a Korean dataset suitable for multi-hop QA, we have taken the initiative to create a dataset specifically tailored for evaluating Korean multi-hop QA. We have meticulously crafted 200 questions that require multi-reasoning capabilities based on the types described in Table 1. These questions are human-generated to specifically address diverse aspects of multi-reasoning, ensuring a comprehensive evaluation of our approach. In creating these questions, we strictly adhered to ethical guidelines and carefully recognized any sensitive information, ensuring the content was appropriate and non-sensitive. Our goal is to release this rigorously curated subset to the public, contributing a valuable resource to the field of Korean multi-hop QA and encouraging further research with practical and applicable evaluation tools.

To further demonstrate the robustness of our method, we additionally extracted a dataset of 1,000 questions from the "Naver Cue:" real-world Korean QA system logs. In this process, we meticulously anonymized the data to not only uphold privacy standards but also to comply with privacy regulations and ethical standards. This careful approach ensures the protection of user privacy while allowing us to validate our method effectively in a real-world context.

4.2 Evaluation Metrics

Metrics such as ROUGE and Disambig-F1, traditionally used in closed-book QA systems for comparing model-predicted answers against ground-truth data (Lin, 2004; Amplayo et al., 2022), are well-suited for end-to-end evaluation where definitive answers exist. However, It is important to note that in real-world applications, the performance of the query planning module cannot be accurate as the correct answers to user queries are often unknown or variable, posing a significant challenge in assessing the module's effectiveness in practical scenarios. Therefore, our focus is on evaluating the query planning process itself, for which we employ the automated metrics proposed in Section 3.4. Additionally, to ensure the reliability of the Query Evaluator, we conduct a human evaluation as described in Section 5.1. This methodological integration ensures a robust and comprehensive assessment of the query planning component.

4.3 Baselines

We compare our proposed method with several established baselines, each representing a unique approach to RAG-based QA systems.

Direct Generation In this approach, a query generation model directly produces a single query from the question, which is then used for document retrieval. This method focuses on achieving results through a single-time retrieval process based on the initial query.

Chain-of-Thought This method involves the model first generating a Chain-of-Thought (CoT) in response to a question before delivering the final answer. It represents a thoughtful, step-wise approach to query generation and information retrieval, as detailed in various works (Wei et al., 2022; Yoran et al., 2023; Liu et al., 2023).

Previous Context Built on the CoT method, 'Previous Context' method follows a multi-step retrieval approach. It triggers retrieval using the previous context as the query. This method, including works like IRCoT (Trivedi et al., 2022), emphasizes the use of ongoing context for progressive information retrieval.

Self-Ask An extension of the CoT prompting, 'Self-Ask' method differs by having the model explicitly formulate the next follow-up question it intends to answer. It uses a search engine to respond to these sub-questions instead of relying solely on the language model. This method is explored in (Jiang et al., 2023).

4.4 Main Results

Method	S. Coh	Ans.	O. Ass	R. Val (%)
Single-time Retrieval				
Direct Generation	6.78	50.15	5.46	58.5
Chain-of-Thought	6.83	54.75	5.69	64.0
Multi-time Retrieval				
Previous Context	7.02	57.07	6.11	66.0
Self-ask	7.01	59.25	6.16	69.5
ToQ (ours)	**7.02**	**60.45**	**6.18**	**74.0**

Table 2: Performance comparison of baseline methods on the dataset of 200 questions requiring multi-hop reasoning. Abbreviations: S. Coh (Semantic Coherence), Ans. (Answerability), O. Ass (Overall Assessment), R. Val (Response Validity). The methods are categorized into Single-time and Multi-time Retrieval.

Method	S. Coh	Ans.	O. Ass	R. Val (%)
Single-time Retrieval				
Direct Generation	6.95	64.94	6.24	83.4
Chain-of-Thought	6.95	66.79	6.33	86.7
Multi-time Retrieval				
Previous Context	7.09	68.78	6.57	86.9
Self-ask	7.09	69.30	6.57	88.0
ToQ (ours)	**7.12**	**69.33**	**6.62**	**89.0**

Table 3: Performance comparison of baseline methods on the dataset of random sampled 1,000 questions.

Comparison with Baselines Our evaluation begins with a focused analysis on a subset of 200 questions specifically requiring multi-hop reasoning, as illustrated in Table 2. In this targeted evaluation, the Tree of Questions (ToQ) method significantly outperforms established baselines, achieving a Response Validity of 74.0% and demonstrating strong scores in Semantic Coherence and Answerability at 7.02 and 60.45, respectively. This superior performance in a complex multi-hop reasoning context underscores the effectiveness of the ToQ framework in handling intricate queries.

Following the targeted analysis on multi-hop reasoning, we extend our evaluation to a broader dataset of 1,000 randomly sampled questions, the performance of which is detailed in Table 3. This comprehensive evaluation demonstrates that the ToQ method consistently maintains its high level of performance across a diverse range of question types and complexities. The ToQ framework's robust and adaptable performance across a wide range of QA scenarios, including both focused multi-hop reasoning and a diverse set of questions, highlights its versatility and reliability. Its consistent efficacy demonstrates the method's ability to accurately address questions of varying complexity and depth.

Performance Analysis by Tree Depth We evaluate the performance of our method by examining the Response Validity at various depths within the tree. Table 4 presents the current performance of response validity at each tree level. The tree is limited to a maximum depth of four levels, focusing on the effectiveness of our approach in decomposing and addressing complex queries. This analysis provides insight into how the depth of reasoning impacts the quality of responses generated by our system.

As we delve deeper into the tree levels, we observe an increase in Response Validity. This improvement can be attributed to the increased specificity and context-awareness in sub-questions at deeper levels, and the more focused information retrieval that accompanies this specificity.

5 Analysis

5.1 Correlation of Query Evaluator with Human Judgment

We focus on assessing the accuracy of the Query Evaluator by comparing its evaluations with those made by human annotators. The goal is to establish the degree of correlation between automated

Tree Depth	Response Validity (%)
Level 1	58.5
Level 2	68.5
Level 3	72.0
Level 4	**74.0**

Table 4: Performance of Response Validity at different levels of the Tree of Questions, showing a clear trend of increasing satisfaction rate with deeper levels.

and human assessments, thereby validating the reliability and credibility of the Query Evaluator's performance in real-world scenarios.

First, to validate our automated system, we use the Inter-Annotator Agreement (IAA) to measure consistency among human annotators. As noted in Appendix C, the high IAA scores indicate a significant agreement, confirming the reliability of our human judgment benchmark. Second, our analysis extends to examining the correlation between each metric component used by the Query Evaluator (such as Semantic Coherence, etc.) and human annotations. The detailed findings, presented in Appendix D, include Pearson's correlation coefficients for each metric. These coefficients, reveal how closely each aspect of the Query Evaluator's assessment aligns with human judgment.

5.2 Improvement in Handling Failures with Tree of Questions

Analyzing the transition from single-time retrieval failures to ToQ success, we observe a significant improvement. Out of 83 failures in single-time retrieval at level-1, 31 questions (37.3%) are successfully addressed using the ToQ approach, with increasing success rates at deeper levels.

ToQ Level	**Resolved Cases**	**Rate (%)**
Level-1 (Initial Failure)	0	0.0
Level-2	20	24.1
Level-3	27	32.5
Level-4	31	37.3

Table 5: Resolution rates of single-time retrieval failures at different levels of the Tree of Questions.

5.3 Qualitative Analysis

We present a qualitative analysis of our ToQ method, focusing on its ability to effectively handle complex queries, as illustrated in Appendix Figure 3. For instance, in the case of the single-time retrieval method applied to the question, *"Recommend a deposit that is advantageous to young people born in 1996. Please tell me that there are no restrictions on the family's wealth,"* the method exhibits limitations in adequately addressing the query's nuances. In contrast, our ToQ method constructs a question tree node corresponding to a bridging case with an additional depth of two levels. This enables the ToQ to generate more appropriate queries for searching, ultimately providing a more accurate and relevant answer to the original question. The qualitative comparison underscores the enhanced capability of the ToQ method in handling complex, multi-faceted questions.

5.4 Error Case Studies

In our analysis, we identify several types of error cases that pose challenges to our Tree of Questions method. These cases shed light on areas where further improvement is needed.

Inability to Decompose Questions Some questions, such as *"Please show me a photo of the Gochon area in 1977,"* cannot be effectively decomposed into simpler queries, leading to a failure in the ToQ process. These types of questions, which are inherently complex and lack a straightforward decomposition path, comprise approximately 10% of the questions in our dataset, indicating a significant area for potential improvement in handling such intricate questions.

Long-tail Questions Even with a successfully generated query, the absence of reliable documents on the search engine can lead to errors. This is common in long-tail questions such as hyper-specific legal questions, inquiries into particular cultural practices, or detailed comparisons of obscure products.

6 Conclusion

In this paper, we introduce advancements in RAG-based QA systems for Korean, focusing on the Tree-of-Question (ToQ) methodology and enhanced query planning. Our evaluations show the ToQ method's effectiveness in multi-hop reasoning and its adaptability across a comprehensive dataset. Notably, ToQ significantly improves handling complex Korean language queries by enabling deeper reasoning. Additionally, we present a novel evaluation method in a detailed Korean multi-hop QA dataset. Our contributions pave the way for more accurate and context-sensitive QA systems, especially for languages with unique challenges like Korean.

Limitations

Language Scope and Future Expansion While our study offers significant insights into multi-hop question answering for the Korean language, leveraging a model specifically designed for Korean, it's important to recognize its limitations in terms of language scope. Our experiments were conducted exclusively on Korean datasets, validating the effectiveness of our method in this specific linguistic context. However, to broaden the applicability and validate the universality of our approach, we plan to extend our experiments to English datasets. This expansion will involve using other Large Language Models as the backbone.

Challenges in Addressing Long-tail Questions Another limitation in our approach arises when dealing with long-tail questions. These questions often pertain to highly specialized or niche topics, such as detailed legal inquiries, specific cultural practices, or comparisons of obscure products. Even if our system successfully generates a query for such questions, the limitation lies in the availability of relevant and reliable documents within the search engine's database. The scarcity of comprehensive information on these niche topics can result in inaccuracies or incomplete answers.

References

Reinald Kim Amplayo, Kellie Webster, Michael Collins, Dipanjan Das, and Shashi Narayan. 2022. Query refinement prompts for closed-book long-form question answering. *arXiv preprint arXiv:2210.17525*.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

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*.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Few-shot reranking for multi-hop qa via language model prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15882–15897.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Dong Hyeon Jeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, et al. 2021. What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. *arXiv preprint arXiv:2109.04650*.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Dongyub Lee, Myeongcheol Shin, Taesun Whang, Seungwoo Cho, Byeongil Ko, Daniel Lee, Eunggyun Kim, and Jaechoon Jo. 2020. Reference and document aware semantic evaluation methods for korean language summarization. *arXiv preprint arXiv:2005.03510*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xiangyang Liu, Tianqi Pang, and Chenyou Fan. 2023. Federated prompting and chain-of-thought reasoning for improving llms answering. *arXiv preprint arXiv:2304.13911*.

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.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2022. A survey on multi-hop question answering and generation. *arXiv preprint arXiv:2204.09140*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. *arXiv preprint arXiv:1906.02916*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: browser-assisted question-answering with human feedback (2021). *URL https://arxiv.org/abs/2112.09332*.

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. An empirical study of tokenization strategies for various korean nlp tasks. *arXiv preprint arXiv:2010.02534*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Yujia Qin, Zihan Cai, Dian Jin, Lan Yan, Shihao Liang, Kunlun Zhu, Yankai Lin, Xu Han, Ning Ding, Huadong Wang, et al. 2023. Webcpm: Interactive web search for chinese long-form question answering. *arXiv preprint arXiv:2305.06849*.

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*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, et al. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*.

Suhyune Son, Chanjun Park, Jungseob Lee, Midan Shim, Chanhee Lee, Kinam Park, and Heuiseok Lim. 2022. Korean and multilingual language models study for cross-lingual post-training (xpt). *Journal of the Korea Convergence Society*, 13(3):77–89.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Kichang Yang. 2021. Transformer-based korean pre-trained language models: A survey on three years of progress. *arXiv preprint arXiv:2112.03014*.

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. *arXiv preprint arXiv:1809.09600*.

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 preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Kang Min Yoo, Jaegeun Han, Sookyo In, Heewon Jeon, Jisu Jeong, Jaewook Kang, and et al. Hyunwook Kim. 2024. Hyperclova x technical report. *arXiv preprint arXiv:2404.01954*.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. *arXiv preprint arXiv:2304.13007*.

Hongyeon Yu, Seung Hak Yu, and Young Bum Kim. 2023. Naver cue: Search service based on large language models. *Communications of the Korean Institute of Information Scientists and Engineers*, 41:34–41.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

A Algorithm of Tree of Questions

B Prompt Examples

Prompt B.1: Answer Integrator

- Search the document for an answer span that exactly matches the intent of the user's question.
(원문의 의도에 정확히 부합하는 답변 범위를 문서에서 찾습니다.)
- If the question and document are relevant, extract the answer span from the document that matches the user's question intent.
(질문과 문서가 관련이 있으면, 사용자의 질문 의도에 맞는 답변 범위를 문서에서 추출합니다.)
- If the question and document are irrelevant, output None.
(질문과 문서가 관련이 없으면, None을 출력합니다.)
- Output in the following format:
(아래 형식으로 출력합니다:)
{"relevance": "relevant | irrelevant", "answer_span": "${relevant span}"}
- The following is an example.
(다음은 예시입니다.)
Symbol of Courage
{"title": "Symbols that symbolize good luck", "summary": "Let's take a look at the various symbols." ..."}
({"title": "행운을 상징하는 상징들", "summary": " 오늘날까지 이어지는 다양한 행운의 상징들을 살펴보겠습니다. ..."})
{"relevance": "irrelevant", "answer_span": "None"}

Prompt B.2: Question Decomposer

- Evaluates whether the user's inquiry can be addressed through a single query in a search engine or whether it requires multiple searches to compile the necessary information.
(당신은 사용자의 질문을 검색 엔진 한 번 검색으로 정보 수집이 가능한 질문인지, 여러 번 검색을 통해 정보를 수집해야하는지 판단합니다.)
- If multiple searches are required, decompose the question into multiple sentences.
(여러 번 검색이 필요한 경우 질문을 여러개의 문장으로 분리합니다.)
- If a single search is required, return the user's question without modification.
(한 번의 검색이 필요한 경우 사용자의 질문을 그대로 출력합니다.)
- If the answer to a previous question needs to be used again as a question, mark it as [ANS_N].
(이전 질문의 답변을 다시 질문으로 활용해야하는 경우 [ANS_N]으로 표시합니다.)
- The following is an example:
(다음은 예시입니다:)
Please recommend an electric car in a similar price range to the BMW i5.
(BMW i5와 유사한 가격대의 전기차를 추천해주세요.)
1. What is the price range of BMW i5?
(1. bmw i5 가격대가 얼마야?)
2. Please recommend an electric car in the price range of [ANS_1].
([ANS_N] 가격대의 전기차 추천해줘)

Prompt B.3: Query Generator

- You are a model that generates queries to search users' questions on search engines.
(검색 엔진에서 사용자의 질문을 검색하는 쿼리를 생성하는 모델입니다.)
- Create one optimal search term to answer your question.
(질문에 대한 답을 찾기 위한 최적의 검색어를 생성합니다.)
- Examples:
Please recommend an electric car in a similar price range to the BMW i5.
(BMW i5와 유사한 가격대의 전기차를 추천해주세요.)
Query: recommendation of BMW i5 price range electric car.
(bmw i5 가격대 전기차 추천.)
Please tell me the Samsung stock price.
(삼성 주식 가격을 알려주세요.)
Query: Samsung stock price (삼성 주식 가격)

Prompt B.4: Query Evaluator

- Evaluates the semantic_coherence and answerability of each summary for the user question.
(사용자 질문에 대한 각 요약의 semantic_coherence와 answerability를 평가합니다.)
- *Semantic Coherence*: Evaluation of how the summary maintains a logical flow and relevance to the user's question. Scores range from 1 (not at all) to 10 (exact match).
(*Semantic Coherence*: 요약이 논리적인 흐름을 유지하고 사용자 질문과 어떻게 관련성을 유지하는지에 대한 평가. 점수는 1(전혀 없음)에서 10(완전 일치)까지입니다.)
- *Answerability*: Estimation of the probability that the summary directly and completely answers the user question. Confidence is expressed as a percentage, with 0% indicating no confidence and 100% indicating complete confidence.
(*Answerability*: 요약이 사용자 질문에 직접적이고 완전하게 답하는 확률을 추정. 신뢰도는 퍼센트로 표시되며, 0%는 답변 가능성에 대한 신뢰가 없음을, 100%는 완전한 신뢰를 의미합니다.)
- Each summary's overall assessment score is calculated by averaging the Semantic Coherence and Answerability results, converting Answerability from a 0%-100% score to a 1-10 scale.
(각 요약에 대한 전체 평가 점수는 Semantic Coherence와 Answerability 결과를 평균하여 계산되며, Answerability는 0%-100% 점수를 1-10 척도로 변환하여 계산합니다.)
- **Examples:**
Why cosmetics review ratings are important
[Cosmetics review rating meaning]: Cosmetics review rating is an indicator that evaluates product quality and user satisfaction. ...
(*화장품 리뷰 평점의 중요성에 대해서*
[화장품 리뷰 평점의 의미]: 화장품 리뷰 평점은 제품 품질과 사용자 만족도를 평가하는 지표입니다. ...)
{"semantic_coherence": 9, "answerability": 95, "overall_assessment": 9.5, "response_validity": true}

C Inter-Annotator Agreement (IAA) Measurements of Query Evaluator

In this section, we present an in-depth analysis of the Inter-Annotator Agreement (IAA) for our Query Evaluator. The IAA is a crucial metric in evaluating the consistency and reliability of human annotators when assessing the outputs generated by our Query Evaluator. It serves as an indicator of the degree to which different annotators provide similar ratings, thereby offering insights into the validity and interpretability of the Query Evaluator's performance.

To conduct this analysis, we engaged five human annotators, authors of this paper, to assess a sample of 100 queries processed by the Query Evaluator. The queries were evaluated based on predefined criteria, with the aim to compare the consistency of the human annotators' judgments. Two distinct IAA (Inter-Annotator Agreement) measurements were employed: the Direct Generation IAA and the Tree-of-Question (ToQ) IAA.

Measurement	PA	PE	Fleiss' Kappa
Direct Generation	0.872	0.500	0.744
Tree-of-Question	0.892	0.588	0.738

Table 6: Inter-Annotator Agreement (IAA) Measurements.

As illustrated in the Table 6, both IAA measurements exhibit substantial levels of agreement among the annotators. In the Direct Generation IAA, the Proportional Agreement (PA) was noted as 0.872, indicating a high level of consensus among annotators in their evaluations. Similarly, the Fleiss' Kappa value of 0.744 in this measurement suggests a substantial agreement beyond chance.

In the ToQ retrieval IAA, there was a slight increase in PA to 0.892, indicating an even higher level of agreement among the annotators for this set of queries. The Fleiss' Kappa value of 0.738, although slightly lower than in the Direct Generation scenario, still indicates a substantial agreement level.

The Probability of Chance Agreement (PE) in both measurements also reflects noteworthy observations. For the Direct Generation IAA, the PE is 0.500288, while for the ToQ retrieval IAA, it is higher at 0.5882. These values indicate that while there is some element of chance agreement, the high Fleiss' Kappa values demonstrate that the majority of the agreement among annotators is due to their consistent judgment rather than chance.

The consistency in these IAA measurements is a testament to the reliability of human annotators in evaluating the queries processed by the Query Evaluator. This consistency also affirms the robustness of the Query Evaluator's output, as it aligns closely with human judgment, which is critical in ensuring the practical applicability of the Query Evaluator in real-world scenarios.

D Correlation Analysis Between Human and Query Evaluator Metrics

In this section, we present the results of our correlation analysis between the consensus annotations from five annotators and the metrics computed by our Query Evaluator. We employ a majority voting system to aggregate the binary (True/False) annotations for each query, resulting in a representative consensus for each. Subsequently, we calculate both Pearson and Spearman correlation coefficients to understand the linear and monotonic relationships, respectively, between these consensus annotations and each metric of the Query Evaluator.

Majority Voting Aggregation To aggregate the annotations, we implement a majority voting mechanism. For each query, we determine the most common annotation (True or False) among the five annotators. This approach allows us to capture the dominant trend in human judgment for each query.

Correlation Coefficient Calculation We calculate the Pearson and Spearman correlation coefficients for the following metrics of the Query Evaluator against the aggregated annotations: 1) Semantic Coherence, 2) Answerability, 3) Overall Assessment Score, and 4) Response Validity.

Each metric is correlated with the consensus annotation to gauge its alignment with human judgment. Pearson correlation was used to assess the linear relationship, while Spearman correlation was employed to understand the rank-order relationship.

Each metric is correlated with the consensus annotation to gauge its alignment with human judgment. Pearson correlation is used to assess the linear relationship, while Spearman correlation is employed to understand the rank-order relationship.

Results of Pearson Correlation Analysis As described in Figure 2, pearson correlation analysis yields the following results:

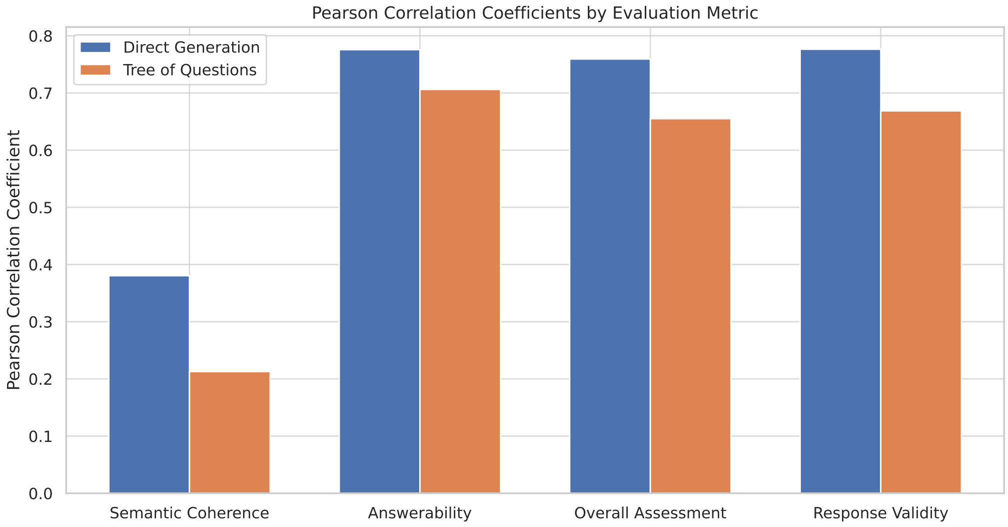
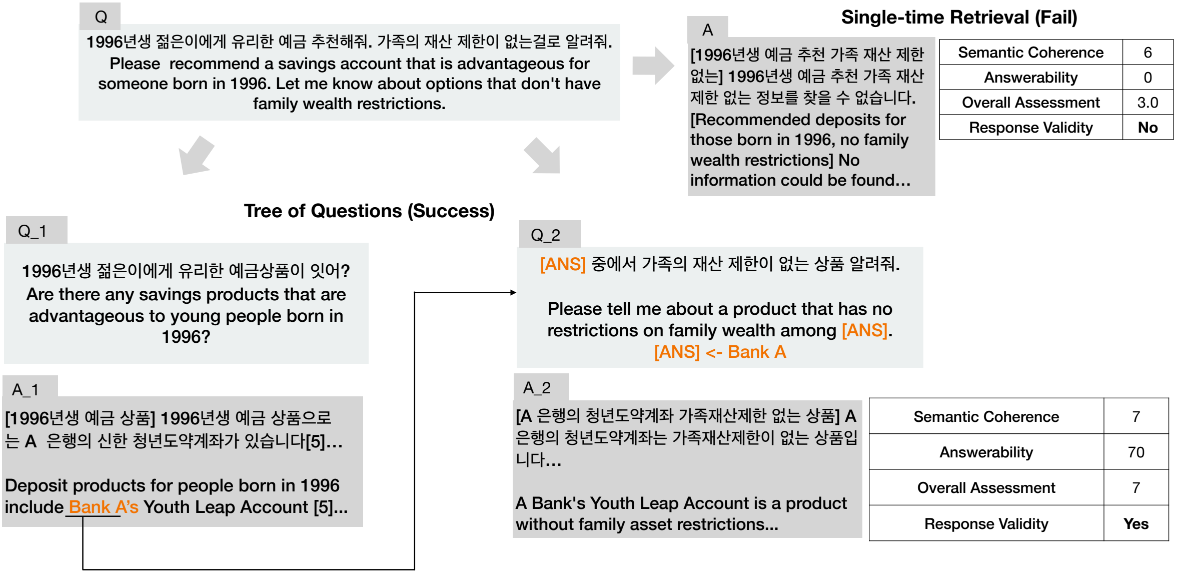


Figure 2: Comparison of Pearson Correlation Coefficients for 'Direct Generation' and 'Tree of Questions' methods, illustrating the distinct performance characteristics of each in terms of Semantic Coherence, Answerability, Overall Assessment, and Response Validity.



1996년생 젊은이에게 유리한 예금 추천해줘. 가족의 재산 제한이 없는걸로 알려줘.
Please recommend a savings account that is advantageous for someone born in 1996. Let me know about options that don't have family wealth restrictions.

[1996년생 예금 추천 가족 재산 제한 없는] 1996년생 예금 추천 가족 재산 제한 없는 정보를 찾을 수 없습니다.
[Recommended deposits for those born in 1996, no family wealth restrictions] No information could be found…

Semantic Coherence	6
Answerability	0
Overall Assessment	3.0
Response Validity	**No**

1996년생 젊은이에게 유리한 예금상품이 잇어?
Are there any savings products that are advantageous to young people born in 1996?

[1996년생 예금 상품] 1996년생 예금 상품으로는 A 은행의 신한 청년도약계좌가 있습니다[5]…

Deposit products for people born in 1996 include Bank A's Youth Leap Account [5]...

[ANS] 중에서 가족의 재산 제한이 없는 상품 알려줘.

Please tell me about a product that has no restrictions on family wealth among [ANS].
[ANS] <- Bank A

[A 은행의 청년도약계좌 가족재산제한 없는 상품] A 은행의 청년도약계좌는 가족재산제한이 없는 상품입니다…

A Bank's Youth Leap Account is a product without family asset restrictions...

Semantic Coherence	7
Answerability	70
Overall Assessment	7
Response Validity	**Yes**

Figure 3: Qualitative example of Tree-of-Questions Framework.

- Semantic Coherence: For the 'Direct Generation' method, the Pearson correlation coefficient is 0.3804, indicating a moderate positive relationship with human annotations. For the 'Tree-of-Question' method, the correlation is lower at 0.2128. The lower correlation of Semantic Coherence compared to other metrics can be attributed to the fact that it tends to achieve some level of coherence by mentioning content related to the user's question, even if the question isn't answered directly. This distribution of scores ranging from 6 to 7 points suggests that the metric may not effectively capture the depth or relevance of the answer to the user's query, as it may assign relatively high scores even when the answer is not fully satisfying in terms of providing a direct response.

- Answerability: The 'Direct Generation' method shows a strong positive correlation of 0.7757, suggesting high agreement with human judgment. The 'Tree-of-Question' method has a correlation of 0.7061.

- Overall Assessment Score: This metric also demonstrates a strong positive correlation for both methods, with 'Direct Generation' at 0.7593 and 'Tree-of-Question' at 0.6550.

- Response Validity: The strongest correlation with human annotations is observed in the 'Response Validity' metric, with 'Direct Generation' at 0.7764 and 'Tree-of-Question' at 0.6686.

These results indicate the overall assessment and response validity are particularly strong indicators of human judgment across both methods.

E Experimental Setup

As outlined in Section 2, our experimental framework assumes the existence of both the *Document Retrieval* and the *Response Generation* in-house models for retrieving documents and generating responses. Our primary focus is on developing an effective Query Planner component. The models employed in the processes described in Sections 3.1, 3.2, 3.3, and 3.4 all utilize the 60B parameter HyperCLOVAX (Kim et al., 2021; Shin et al., 2022) as their backbone large language model.

In our setup, the *Document Retrieval* model, functioning as Naver's in-house search engine, retrieves three related documents based on the query generated through the Tree-of-Questions (ToQ) and the Query Generator as discussed in Sections 3.1 and 3.3, respectively, from a question. Subsequently, the *Response Generation* model processes these documents to generate the final response, denoted as R.

F Related Work

Initial advancements in long-form complex question answering (QA) based on large language models have leveraged the Chain-of-Thought (CoT) approach (Wei et al., 2022). Attempts to enhance the performance of QA models through sophisticated prompting techniques have set the stage for further developments in this area (Sun et al., 2022; Lazaridou et al., 2022; Yu et al., 2022; Khalifa et al., 2023). Building on this foundation, recent efforts have increasingly focused on utilizing retrieval-based approaches. These efforts aim to augment the factual knowledge inherent in LLMs with retrieval search results (Nakano et al.; Mallen et al., 2023; Qin et al., 2023). Despite the significant progress made, these methods often face challenges in scenarios requiring multiple active retrievals.

In response to these challenges, research has shifted towards developing multi-time retrieval methods. A notable method in this category is retrieving additional information using previous context at predetermined intervals (Khandelwal et al., 2019; Borgeaud et al., 2022; Ram et al., 2023). However, these methods can be inefficient due to their reliance on previously generated tokens for queries and the fixed nature of the retrieval intervals.

Another significant approach in the field of multi-time retrieval for QA involves decomposing comprehensive questions into smaller, more manageable sub-questions, which aids in targeted information retrieval (Yao et al., 2022; Khot et al., 2022; Khattab et al., 2022; Press et al., 2022; Jiang et al., 2023). This strategy has shown increased efficiency in determining the timing of retrievals, leveraging the inherent knowledge of LLMs.

However, as highlighted by Huang et al. (2023), relying solely on the inherent reasoning capabilities of LLMs without external feedback can lead to performance degradation. Our study addresses this issue by focusing on the generation of queries within a RAG-based multi-hop reasoning QA system. Therefore, we propose an interactive and explicit evaluation method that assesses whether the queries generated are sufficient to answer user questions, thus ensuring the creation of more effective and reliable responses.