FaVe: Factored and Verified Search Rationale for Long-form Answer

Jihyuk Kim[*]
LG AI Research
jihyuk.kim@lgresearch.ai

Sungjin Lee
Samsung
sj21.lee@samsung.com

Seung-won Hwang[†]
Seoul National University
seungwonh@snu.ac.kr

Yang Liu
Amazon Alexa AI
yangliud@amazon.com

Abstract

Targeting long-form question-answering, chain-of-query (CoQ) has been studied, integrating chain-of-thought (CoT) with retrieval-augmented generation. CoQ breaks down complex questions into simpler subquestions (SQs), allowing relevant information to be retrieved step by step. By doing so, CoQ aims to improve the answer comprehensiveness and verifiability, at the expense of latency. Our first contribution is showing that the chaining often incurs harmful effects on both objectives, and SQs left unverified often fail to answer the given question. Second, we propose a better alternative to CoQ, *union-of-query* which adopts a factored approach to break the harmful chain. Finally, we propose to verify SQs before answers, by fine-tuning the SQ generator using verified SQs and introducing a selector verifying SQs in test time. Employing `vicuna-13b`, our approach, denoted by **FaVe** (short for Factored and Verified search), even outperforms ChatGPT baselines while maintaining efficiency.

1 Introduction

In long-form question-answering (LFQA) (Fan et al., 2019), a model is tasked to generate a long-form response to answer a complex question. In this work, we target two task objectives for LFQA: answer comprehensiveness and answer verifiability (Gao et al., 2023b), where we aim to generate factual claims that are not only comprehensive, meaning they cover all relevant information related to the given question, but also verifiable, ensuring that the claims can be confirmed through external sources (Rashkin et al., 2023).

An existing solution is retrieval-augmented generation (RAG) (Lewis et al., 2020; Izacard et al., 2022), utilizing retrieved documents from external corpus as *rationales* for the generated answer. This enables the user to verify the truthfulness of the answer via citations to the documents (Nakano et al., 2021; Menick et al., 2022). Extending it beyond single-turn querying, chain-of-query prompt-

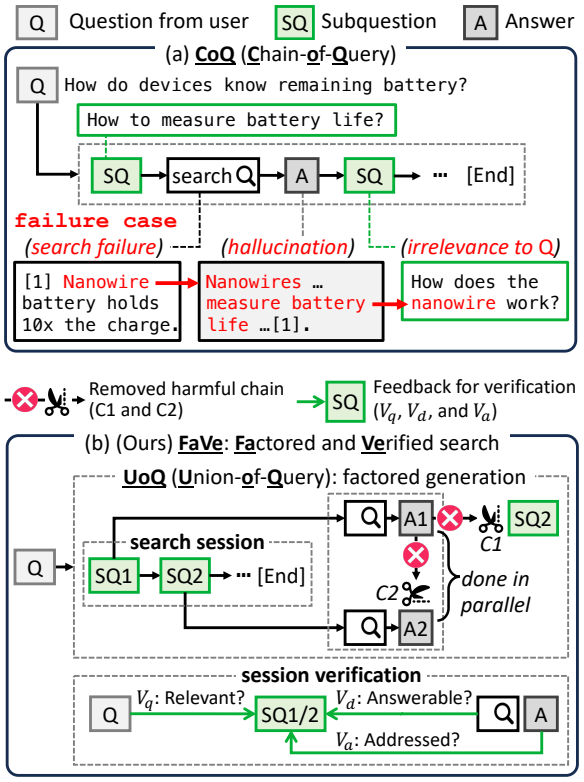

Figure 1: Illustration of (a) CoQ and its failure case, and (b) our proposed solution, FaVe.

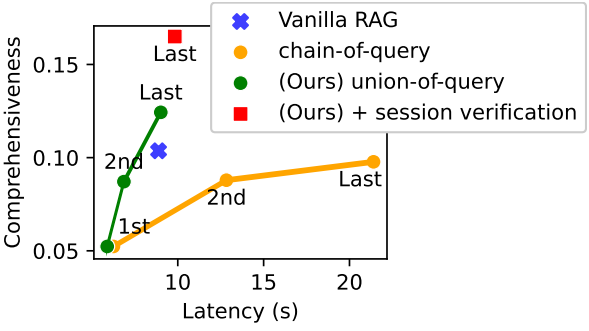

Figure 2: The answer comprehensiveness (y-axis) and latency (x-axis) at different search iterations.

[*]Work done during internship at Amazon Alexa AI.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 19402–19416
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

ing (CoQ) (Yao et al., 2023; Jiang et al., 2023; Shao et al., 2023; Press et al., 2023) has been studied for improved comprehensiveness. As depicted in Figure 1(a), CoQ begins with breaking down a complex question into a simpler subquestion (SQ) and an answer for it, and then, if any relevant claim remains unaddressed, continues to explore it in subsequent iterations, conditioned by previous ones, i.e., forming *a chain of queries*. At the expense of the latency cost, such multi-turn querying is intended to achieve progressively more comprehensive answers through successive search iterations.

However, Figure 2 shows otherwise[1]: Presented by orange curve, comprehensiveness quickly saturates, showing little improvements over vanilla RAG (blue X mark), while latency surpasses it in just two iterations. As attributions for the degeneration, we identify two limitations of CoQ, illustrated in Figure 1(a) with a failure case (red-colored texts): 1) The chain may propagate errors from previous answers to subsequent generations, thereby becoming rather a *harmful chain*, and 2) *unverified SQ*s often fail to answer the given question. Figure 1(b) presents our solution to tackle the two limitations.

First, we propose **union-of-query** (**UoQ**) adopting a factored approach, to break two harmful Chains (as shown in Figure 1(b) with scissors) – (C1) the chain between answers and queries and (C2) the chain between different answers – while preserving the chain between SQs. By breaking C1, UoQ, before answers, sequentially decodes varying numbers of SQs, i.e., a *search session*. Then, by breaking C2, UoQ answers different SQs in parallel, reducing latency. Figure 2 shows that UoQ (green curve) via the chain break consistently improves the comprehensiveness and maintains comparable latency to that of vanilla RAG.

Second, we propose **session verification**, leveraging feedback presented in Figure 1(b) with green arrows, to Verify if an SQ is (V_q) relevant to the given question, (V_d) answerable by the retrieved document, and finally (V_a) addressing relevant claims in the final answer. Specifically, the verification process comprises generating multiple search session candidates and, to identify the best session, verifying SQs in each session candidate based on the received feedbacks. Building upon this process, we introduce *answer-aware* candidate generator, which is trained using verified sessions via feedback for V_a, and *unified* session selector, which verifies SQs with the unified feedback encompassing V_q and V_d to select the best session among the candidates. Figure 2 shows that the session verification further improves performance with little sacrifice on latency (red square).

With these two components (i.e., UoQ and session verification), we propose Factored and Verified search session, or **FaVe**. We evaluate models using two datasets: ELI5 (Fan et al., 2019) and StrategyQA (Geva et al., 2021). Compared to CoQ, results on ELI5 show that FaVe improves answer comprehensiveness and verifiability by 35.4% and 21.8%, respectively, while reducing latency by a factor of two via parallel answer generation. When evaluated on multi-hop reasoning using StrategyQA, FaVe also demonstrates improved coverage in relevant facts by 35.6%, achieving 2.9%pt increases in the final accuracy on Yes/No questions.

Our contributions are threefold:

1. We identify that the presence of two harmful chains of CoQ results in adverse effects, accompanied by considerable latency costs. To tackle this, we propose UoQ, which shows better performance with lower latency by breaking these chains.

2. We identify three failure cases of SQs when left unverified. To address each of these cases, we propose tailored feedback, which is utilized by the answer-aware session generator and the unified session selector. In addition, our unified selector can be easily integrated with CoQ, to enable error-free chaining if high latency costs are acceptable.

3. We thoroughly assess the effectiveness of FaVe, using both ELI5 and StrategyQA which mainly target multifaceted or interdependent facts, respectively, through extensive evaluations conducted by both humans and automated methods.

2 Related Work and Motivation

Though large language models (LLMs) often produce factually incorrect answers, grounding knowledge or latent rationales of answers are unknown to users, constraining reliability of LLM-generated responses. To enhance reliability, rationales can be obtained 1) internally or 2) externally.

A notable exemplar of the former is chain-of-thought prompting (Wei et al., 2022), where LLM

[1]The comprehensiveness is measured by gold claim recall, used in ALCE benchmark (Gao et al., 2023b) on ELI5 dataset (Fan et al., 2019).

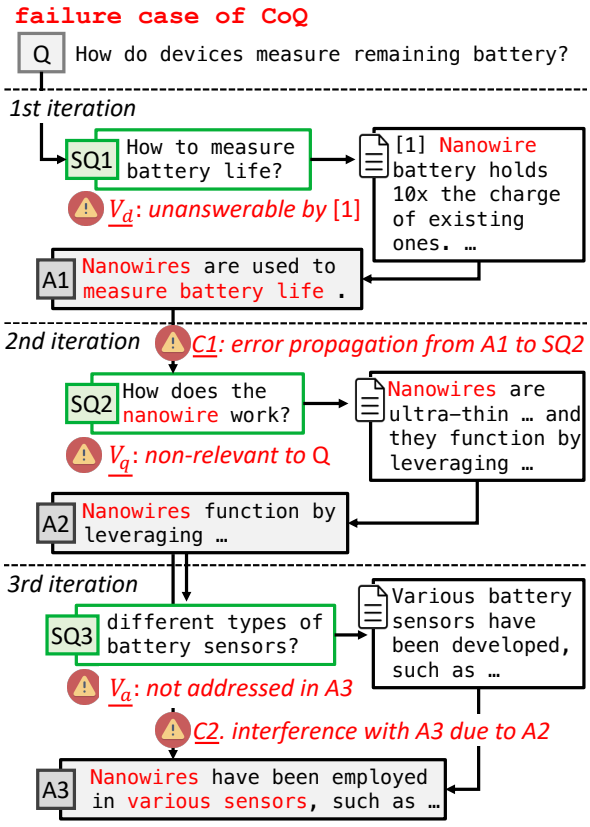

Figure 3: A failure case of CoQ illustrating two harmful chains (C1 and C2) and three negative influences of unverified SQs (V_d, V_q, and V_a).

produces a reasoning chain that serves as the underlying logic behind its answer. CoT has shown promising results in reasoning tasks (Kojima et al., 2022; Wang et al., 2023). On the other hand, for knowledge-intensive tasks (e.g., LFQA), rationale based on external knowledge becomes more appealing, which offers complementary knowledge to LLMs' internal knowledge (Lewis et al., 2020; Izacard et al., 2022). The user can verify the truthfulness of the answer via citations to the documents retrieved from external corpora (Rashkin et al., 2023).

Recently, "chain-of-query" prompting has gained attention, as it combines both internal and external grounding. Given a complex question, CoQ generates an SQ through internal reasoning, answers the SQ by retrieving external rationale, and iteratively repeats these two processes (Yao et al., 2023; Jiang et al., 2023; Shao et al., 2023; Press et al., 2023) (Prompts are shown in Appendix A.1.). Despite the potential, CoQ showed limited improvements in LFQA tasks (Gao et al., 2023b).

Motivation Our distinction is to uncover the two key issues in CoQ: *harmful chaining* and *unverified subquestions*. Figure 3 showcases two harmful chains (C1/2) and three negative influences of SQs left unverified ($V_{[d/q/a]}$): Given the question Q, CoQ begins with generating SQ1. Despite the relevance to Q, (V_d) SQ1 is not answerable by the retrieved document, producing hallucinations in A1. Nevertheless, in the second iteration, (C1) the error in A1 propagates to the subsequent SQ, producing (V_q) SQ2 non-relevant to Q, such that A2, as well, becomes non-relevant. For the last iteration, SQ3, though relevant to Q and answerable by the document, (V_a) is not faithfully addressed in A3, due to (C2) the interference from A2.

Inspired by the failure case, we propose 1) **UoQ** breaking the two harmful chains (C1/2) while retaining the chain between SQs to progressively explore comprehensive knowledge, and 2) introduce **session verification** to avoid the three negative influences of unverified SQs ($V_{[d,q,a]}$).

Similar to UoQ, prior work has employed factored generation for revising initial answers (Gao et al., 2023a; Dhuliawala et al., 2023). In contrast, we focus on generating answers from scratch. While the initial answer offers a comprehensive set of relevant claims, our distinction is progressively achieving it via multiple SQs. To this end, though CoQ has been assumed effective, our contribution lies in challenging and refuting the assumption and proposing UoQ as a superior alternative. Furthermore, SQs, as generated from scratch, often lose relevance to the given question, exacerbated by search failures. Our work is further distinguished by proposing novel verification methods to tackle this.

Regarding the verification, while standard approaches often assume the availability of straightforward ground-truth checks, such as exact matches with gold numeric answers in math tasks (Snell et al., 2024), or solely rely on the LLM's internal feedback, referred to as "self-verification" (Weng et al., 2023; Dhuliawala et al., 2023), our distinction is incorporating external feedback as well, such as search engine feedback to ensure the quality of search results (§3.2). Meanwhile, as another dimension of verification, RR (He et al., 2022) *verifies answers* by evaluating whether each answer is faithful to retrieved documents. Our distinction is *verifying SQs*, which is orthogonal and complementary to the answer verification. SQs act as cues for what to answer, such that the quality of SQs largely influences the upper-bound quality of the

Algorithm 1 FaVe for generating answers to the question

Input: The given question q from a user

1: **Search session generation**: Sample multiple search sessions, with each search session s comprising a sequence of N sub-questions of q, that is, $s \leftarrow \{\tilde{q}_1, \ldots, \tilde{q}_i, \ldots, \tilde{q}_N\}$
2: **Search**: For each search session s, search for relevant documents using each subquestion, in parallel, producing top-M documents, $\{d_{i,1}, \ldots, d_{i,m}, \ldots, d_{i,M}\}$ for each $\tilde{q}_i$.
3: **Test-time search session verification**: Evaluate each s regarding its relevance to q (denoted by r_q) and its answerability from $\{d_{i,m}\}_{\forall i \in [1,N], m \in [1,M]}$ (r_d), producing $r \leftarrow (r_q + r_d)/2$.
4: **Factored answer generation**: For the subquestions in the best s with the highest r, generate answers to each subquestion, in parallel, based on the top-M documents, producing $a \leftarrow \{\tilde{a}_1, \ldots, \tilde{a}_i, \ldots, \tilde{a}_N\}$. Optionally, post-processing on a can be followed to conclude the final prediction.

resulting answers (Deng et al., 2023).

3 Approach

In the following sections, we first describe UoQ, the factored generation process targeting a single search session (§3.1), and then present the session verification process, sampling and verifying multiple search session candidates (§3.2). The algorithm for our proposed method is presented in Algorithm 1. For all inferences, such as session generation and answer generation, we employ `vicuna-13b` with prompting, unless specified otherwise. Detailed prompts are shown in Appendix A.2.

3.1 Union-of-Query

UoQ involves three stages: 1) search session generation factoring out answers from SQ generation, 2) document retrieval, and 3) factored answer generation. We elaborate on the three stages below.

Search session generation We prompt LLM to generate a sequence of varying numbers of SQs (which will be verified later (§3.2)), denoted by $s = \{\tilde{q}_1, \cdots, \tilde{q}_i, \cdots, \tilde{q}_N\}$ where N denotes the total number of SQs. Factoring out answers from generating SQs, UoQ improves comprehensiveness, by encouraging SQs to explore diverse knowledge, before delving into detailed and possibly incorrect answers, as shown via C1 in Figure 3.

Document retrieval Employing SQs as search queries, we search relevant documents from a Web corpus, more precisely Sphere (Piktus et al., 2021) or Wikipedia, for ELI5 and StrategyQA, respectively, following each benchmark setting. For efficiency, we first filter candidate documents by retrieving the top-100 documents using BM25 search (Robertson and Walker, 1994) and the given question from the user as the query. Given the candidate documents, we rerank the documents using cross-encoder (Nogueira et al., 2020)[2] and $\tilde{q}_i$ as the query[3]. We take the top-M documents, denoted by $\{d_{i,m}\}_{\forall i \in [1,N], m \in [1,M]}$ where m denotes the rank of each document. We set M by 2 in our experiments. The documents serve as rationales for answers, ensuring their verifiability.

Factored answer generation Finally, UoQ generates the answer $\tilde{a}_i$ for each $\tilde{q}_i$, conditioned only by the target rationale $\{d_{i,1}, \cdots, d_{i,M}\}$, not by previously generated answers $\{\tilde{a}_j\}_{j<i}$[4]. The factored answer generation improves the answer verifiability, by encouraging the answer to be more faithful to the designated rationale, and avoiding distraction between answers as shown via C2 in Figure 3.

To answer each SQ, we employ the same prompt used for CoQ, yet without involving previous answers and SQs. This enables parallel prompting for different SQs, significantly reducing latency. The final answer a is obtained by concatenating the individual answers $\{\tilde{a}_1, \cdots, \tilde{a}_i, \cdots, \tilde{a}_N\}$. For questions requiring concise final answers, such as numbers or entities, we additionally prompt LLM to generate the final prediction using a as input.

Meanwhile, SQs, when left unverified, often fail to provide satisfactory answers to the given question as showcased in Figure 3 via SQ1-3. To answer from verified SQs, we propose session verification, as detailed below.

3.2 Session verification

Our objective in session verification is threefold: to ascertain that SQs in the search session are 1) relevant to the given question, 2) answerable via retrieved documents, and 3) faithfully addressed in the answer with relevant factual claims, for which we introduce tailored feedback as follows.

Standard self-verification approaches (Weng et al., 2023) often only address the first objective

[2] https://huggingface.co/castorini/monot5-base-msmarco

[3] In Appendix E, we compare results from different search methods.

[4] Note that, in contrast to factored answer generation, we sequentially generate SQs, each $\tilde{q}_i$ conditioned on previously generated SQs $\{\tilde{q}_j\}_{j<i}$. Since SQs act as cues for what to answer, such design enables $\tilde{a}_i$ to discuss diverse sub-topics without duplicates for multifacet questions and to make them consistent for interdependent questions.

via the LLM's **internal feedback**: The same LLM, used for generating SQs, inversely checks whether the SQs are relevant to the given question from which it has been generated. However, despite their relevance to the given question, SQs may not be answered by retrieved documents or not addressed in answers, violating the latter two objectives. Our work is distinguished by additionally introducing two external feedbacks, on documents and answers, respectively.

- First, **search engine feedback** examines the answerability of SQs by the retrieved documents. The primary aim is to ensure answer verifiability, as SQs that cannot be answered by the documents would make the answer unfaithful to the documents. Moreover, when combined with the internal feedback, it enhances comprehensiveness by ensuring the inclusion of relevant claims in the document.

- Second, **answer feedback** examines the comprehensiveness of the final answer produced from a search session. To do so, an external natural language inference (NLI) model is employed, examining how many gold claims in the reference answer are entailed by the generated answer. Given that the gold answer is unavailable during testing, we utilize the answer feedback for training.

To accomplish our objectives through feedback, we propose two components: 1) **answer-aware** session generator and 2) **unified** session selector. These components are designed to incorporate two types of feedback – search engine feedback and answer feedback – into the system. Specifically, the answer-aware session generator is trained to generate session candidates that are verified to answer comprehensively via **answer feedback**. In contrast, the unified session selector **unifies** search engine feedback with internal feedback to select the best session. In the subsequent sections, we elaborate on the two proposed components and their roles in leveraging feedback to enhance system performance.

3.2.1 Answer-aware session generator

To produce high-quality session candidates, we fine-tune the session generator, by sampling and verifying multiple sessions for training questions and utilizing those as training examples.

For verifying training data, we utilize the answer feedback based on gold claims in human-annotated answers. Through the answer feedback, we evaluate the comprehensiveness of generated answers rather than their verifiability, as the latter is heavily influenced by an external factor, namely the search engine (which will be addressed later by introducing the selector). To measure the comprehensiveness of each generated answer, we follow the method proposed by Gao et al. (2023b) (§4.2). Specifically, we employ an external NLI model, to count the number of gold claims entailed by the generated answer, as the session comprehensiveness measure.

Once verifying session samples, we employ *contrastive objective* (Lee et al., 2021) for fine-tuning, aiming to prioritize sampling of more comprehensive sessions over less comprehensive ones. Specifically, we compare pairs of sessions and label the one with at least Δ additional factual claims (than the other) as positive and the other as negative, denoted by s^+ and s^-, respectively. Δ was set by two in our experiments, which produces a sufficient number of training pairs with a clear contrast between s^+ and s^-. For collecting such pairs, we used the ELI5 training dataset [5].

With the identified training session pairs in hand, we proceed to optimize the generator using a method called Direct Preference Optimization (DPO) (Rafailov et al., 2023). The generator is optimized to maximize

$$\beta \left(\log \frac{\pi_\theta(s^+)}{\pi_\theta(s^-)} - \log \frac{\pi_{\text{ref}}(s^+)}{\pi_{\text{ref}}(s^-)} \right), \qquad (1)$$

where π_θ and π_{ref} denote the likelihood from the learning generator and reference generator, respectively, and β serves as a regularization term toward the reference model. For π_{ref}, we employ `vicuna-13b` with the prompt in Figure 9 and fine-tune it to produce π_θ, with $\beta = 0.3$, using RMSProp optimizer (Tieleman and Hinton, 2012) with learning rate 5e-7. The optimal checkpoint was set based on the accuracy, $\mathbb{1}(\pi_\theta(s^+) > \pi_\theta(s^-))$, on 1k validation examples.

During testing, we employ the generator to sample multiple session candidates[6], from which we select the best session as explained below.

[5] More precisely, we used 100k questions in the ELI5 training dataset, sampled 8 sessions per question and finally obtained 40k session pairs for contrastive learning.

[6] During testing, we sample 8 search sessions in our experiments for each given question, using top-k sampling with temperature 1.

3.2.2 Unified session selector

In addition to leveraging the most comprehensive session, our goal for the selector is to improve answer verifiability to complement the generator. To achieve this, we unify internal feedback and search engine feedback, to ensure that not only the selected session produces relevant claims to the given question but also their verifiability via the retrieved documents. For each session candidate, the selector first collects internal and search engine feedback on individual SQs, combines the feedback, and computes the final session quality, based on which the best session will be selected.

For the internal feedback, we prompt LLM to conversely examine the relevance of each SQ, $\tilde{q}_i$, to the given question, producing $\{r_i^q\}_{\forall i \in [1,N]}$ where r_i^q is set by 1 if $\tilde{q}_i$ is classified as relevant and 0 otherwise. For the search engine feedback to assess the answerability of SQs, we leverage the relevance scores given by the search engine for the top-1 ranked document, denoted by $\{r_i^d\}_{\forall i \in [1,N]}$. High values for both r_i^q and r_i^d indicate that $\tilde{q}_i$ is relevant to the given question and can be answered using the documents, and thus will offer relevant claims that are grounded in the rationale documents.

Given $\{r_i^q\}_{\forall i \in [1,N]}$ and $\{r_i^d\}_{\forall i \in [1,N]}$ for different SQs in the session, we aggregate scores to measure the final session quality r:

$$r = (r^q + r^d)/2 \in (0, 1) \tag{2}$$

$$r^q = \min\left(\sum_{i=1}^{N} r_i^q, \delta\right)/\delta \in [0, 1] \tag{3}$$

$$r^d = \sum_{i=1}^{N} r_i^d / N \in (0, 1). \tag{4}$$

We introduce δ, which is the predefined maximum number of relevant SQs, to avoid the overgeneration of duplicate gold claims. We set δ by 5, as the answer often contains five distinct factual claims. Given r on each of the session candidates, we leverage the session with the highest r as the rationales for the final answer.

4 Experiment

4.1 Dataset

Two datasets, ELI5 and StrategyQA, were used for evaluation. Both require a model to comprehensively discuss relevant facts to answer a given question, while the two differ in that ELI5 and StrategyQA mainly target multifaceted facts and interdependent facts, respectively.

Specifically, for ELI5, we used 1000 question-answer pairs open-sourced by ALCE (Gao et al., 2023b), which are sub-sampled from ELI5 evaluation dataset. Most of the questions are "why", "how", and "what" questions. These questions demand long-form answers with a length spanning 121.5 words on average. An example of multi-faceted relevant subtopics is presented in Figure 9.

On the other hand, StrategyQA, consisting of 299 Yes/No questions, evaluates models on the multi-hop reasoning task. Given a question (e.g., "Would someone in Mumbai refer to Solanum melongena as an eggplant?"), a model is tasked to produce a long-form answer by reasoning interdependent subquestions and their *answers* (e.g., "In what country Mumbai located? *India*", then, "In what region is India located? *South Asia*", and finally "What is Solanum melongena referred to as in South Asia? *brinjal*"), to finally produce the binary output (e.g., "No" for the exemplar question).

4.2 Evaluation metrics

To evaluate the comprehensiveness and verifiability of long-form answers, we adopt automatic measures proposed by Gao et al. (2023b)[7].

- **Comprehensiveness** is measured by claim recall, which evaluates the ratio of gold claims entailed by the generated answer to the total number of gold claims. We used annotated gold claims in each benchmark dataset.
- **Verifiability** is measured by citation recall, which evaluates the ratio of claims in the generated answer that can be entailed by the citation documents to the total number of claims present in the answer[8]. For assessing the entailment, state-of-the-art NLI model TRUE (Honovich et al., 2022) was employed.

We compared FaVe against vanilla RAG (which uses single-turn querying) and CoQ. For CoQ and our model, we used top 2 documents for each SQ. Vanilla RAG used top 3 documents retrieved from the given question, which performed better in our preliminary study than top 2 or top 4.

[7] In addition to the automatic evaluation, we present human evaluation results in Appendix B, where human annotators prefer FaVe more often than baselines.

[8] Each sentence, split using NLTK tokenizer, is considered to have a single claim.

Model (vicuna-13b)	Claim Recall	Citation Recall	Latency (s)
Vanilla RAG	10.4	45.4	8.9 (s)
CoQ	11.3	45.8	21.9 (s)
(Ours) FaVe	**15.3**	**55.8**	9.1 (s)
w/o $V_{Generator}$	14.0	54.6	9.1 (s)
w/o $V_{Selector}$	13.0	52.6	9.1 (s)
w/o both	12.8	52.1	8.1 (s)

Table 1: The best results are denoted in **bold**, and underlining indicates superior performance compared to CoQ.

4.3 Results on ELI5

Based on the evaluation metrics, in the following sections, we first compare the overall performance on ELI5, to validate the effectiveness of FaVe. Subsequently, we discuss how the two proposed methods, union-of-query and session verification, contribute to improving the task objectives.

UoQ and FaVe outperform CoQ, in terms of generating relevant and verifiable claims, and efficiency as well. Table 1 reports overall performance, along with latency. CoQ shows marginal improvements compared to vanilla RAG, yet hugely sacrifices latency. In contrast, FaVe outperforms all baselines on both metrics and shows much lower latency than CoQ via parallel answer generation. In Appendix C, we present a detailed latency analysis comparing the latency at different stages.

As ablation studies, we excluded fine-tuning the answer-aware session generator (i.e., using LLM without fine-tuning) or the unified session selector for verification, denoted by $V_{Generator}$ and $V_{Selector}$, respectively. Both ablation models decrease the performance, validating the contribution of the two components. Nevertheless, we stress that FaVe without both verifications, i.e., UoQ, still outperforms CoQ, indicating the contribution of chain break.

Table 2 further compares performance using ChatGPT (`gpt-3.5-turbo-0301`). FaVe with `vicuna-13b` even outperforms ChatGPT baselines on both metrics, suggesting that FaVe is more effective than increasing the model size, especially when efficiency matters. Employing ChatGPT for FaVe further improves performance.

[9]For FaVe in Table 2, we excluded the answer-aware session generator, as fine-tuning ChatGPT is infeasible. We also exclude r^q, such that $r = r^d$. When using ChatGPT, more capable of generation, we found that most of the generated SQs are relevant to the given question, such that the internal feedback r^q becomes not needed.

Model	LLM	Claim Recall	Citation Recall
Vanilla RAG	ChatGPT	12.0	51.1
CoQ	ChatGPT	13.4	45.6
(Ours) FaVe[9]	`vicuna-13b`	14.0	54.6
(Ours) FaVe[9]	ChatGPT	**16.3**	**55.5**

Table 2: Evaluation using either `vicuna-13b` or `gpt-3.5-turbo-0301` as backbone LLMs. The **bold**-faced and the underlined denote the best and the second best performance, respectively.

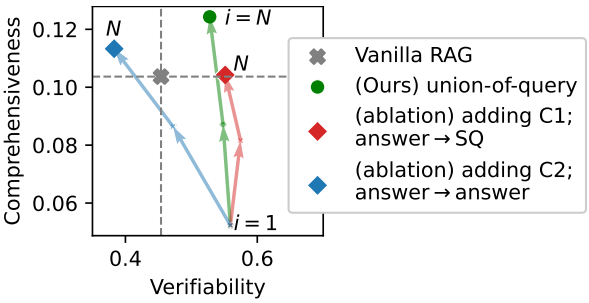

Figure 4: The answer comprehensiveness (y-axis; claim recall) and the verifiability (x-axis; citation recall), from $\tilde{q}_i$ with $i \in [1, 2, N]$.

In the following paragraphs, as in-depth analyses, we validate the effectiveness of each of our two proposed modules, UoQ and session verification.

UoQ outperforms vanilla RAG on both task objectives, while the two harmful chains degrade either of the two. We illustrate how each of the two harmful chains of CoQ (C1 and C2 in Figure 3) adversely affects the two task objectives through ablation studies. UoQ avoids the adverse effects by breaking both chains. Figure 4 presents ablation studies, comparing UoQ with two ablation models that restore each of the two harmful chains indicated by scissors in Figure 1(b), respectively. We report performance on two task objectives, when chaining different numbers of queries.

First, when adding C1 (red curve), the comprehensiveness (y-axis) shows little difference from vanilla RAG. Second, when adding C2 (blue curve), the verifiability (x-axis) significantly decreases, as iteration progresses, at last, falling below that of vanilla RAG[10]. Finally, breaking the two chains,

[10]In Appendix G, we analyze attributions of performance degradation from the perspective of retrieval.

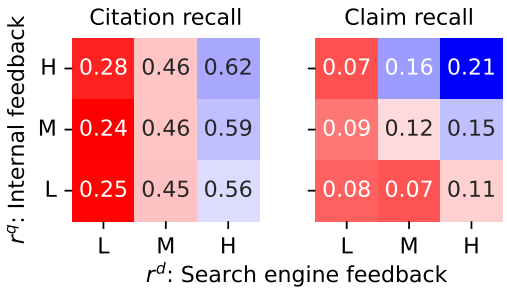

Figure 5: Impact of internal and search engine feedback. The number in each cell denotes the performance from search sessions having r^q and r^d values. Blue/red-colored cell denotes that the performance is better or worse than that of a randomly sampled session (i.e., unverified session). "L", "M", and "H" denote low, medium, and high, respectively.

Model (vicuna-13b)	Claim Recall	Citation Recall	Final Accuracy
Vanilla RAG	$8.1_{\pm 0.6}$	$34.6_{\pm 1.5}$	$51.8_{\pm 2.3}$
CoQ	$13.5_{\pm 1.7}$	$44.1_{\pm 1.9}$	$57.5_{\pm 3.7}$
(Ours) UoQ	$16.9_{\pm 1.1}$	$47.9_{\pm 2.2}$	$58.3_{\pm 1.3}$
(Ours) FaVe[12]	$\mathbf{18.3}_{\pm 1.2}$	$\mathbf{54.5}_{\pm 0.6}$	$\mathbf{60.4}_{\pm 2.8}$
CoQ extension (x3 latency compared to FaVe)			
(Ours) w/ V_{Selector}	$\mathbf{20.8}_{\pm 0.9}$	$\mathbf{50.6}_{\pm 1.2}$	$\mathbf{61.7}_{\pm 2.6}$
w/ self-verification	$19.7_{\pm 0.7}$	$46.7_{\pm 1.4}$	$60.9_{\pm 1.6}$

Table 3: Average performance$_{\pm \text{std}}$ on StrategyQA from 5 different random seeds.

UoQ (green curve) is the only model that outperforms vanilla RAG on both objectives.

Search engine feedback improves answer verifiability, and also the comprehensiveness when jointly used with internal feedback. In Figure 5, we compare internal feedback and external search engine feedback, through the performance of session samples having different r^q and r^d.

The feedback from the search engine r^d is predictive of the citation recall, while the internal feedback r^q is not: When the session has higher r^d, it shows high citation recall regardless of r^q. On the other hand, the claim recall performance is jointly influenced by both feedbacks: The highest claim recall is achieved when both are high, while the performance degrades if either of the two decreases, showing the complementary benefits between the two feedbacks[11].

4.4 Results on StrategyQA

For multi-hop reasoning, CoQ is expected to become a suitable choice, as it extends CoT and *explicitly* reasons interdependent relevant facts step-by-step. On the other hand, UoQ can be considered *implicitly* performing the multi-hop reasoning during the search session generation, by internally envisioning an answer to a subquestion before generating the subsequent subquestion.

In the following paragraphs, on StrategyQA, we first present overall performance results that refute the expectation on CoQ and suggest UoQ and FaVe as promising alternatives, followed by analyzing the limitation and the potential of CoQ. In Appendix D, we present evaluation results on another multi-hop reasoning dataset, HotpotQA (Yang et al., 2018), demonstrating the generalization ability of our method.

UoQ with implicit reasoning outperforms CoQ, and FaVe further enhances UoQ. Given the relatively small number of test questions, we run each model with 5 different random seeds and report the average performance in Table 3. In addition to comprehensiveness and verifiability, we evaluate models on the final accuracy of Yes/No questions.

As expected, when performing the multi-hop reasoning task, CoQ hugely outperforms vanilla RAG, by explicitly reasoning multiple subquestions. Nevertheless, UoQ with implicit reasoning outperforms CoQ and is more efficient. Finally, FaVe, by verifying reasoning steps, further enhances UoQ and outperforms all baselines for all metrics.

In the subsequent paragraph, we further analyze such results from the perspective of error propagation which is an inherent limitation of CoQ. Then, motivated by the analysis, we will also discuss how our unified verification feedback can be applied to CoQ, to overcome its limitation, denoted by "CoQ extension w/ V_{Selector}" in Table 3.

Our unified verification feedback realizes the potential of CoQ, by avoiding errors. Figure 6(a) illustrates our motivation for the CoQ extension, by presenting the limitation and the potential of CoQ: When the model fails to produce gold fact at the first iteration (denoted by "Failure" in the x-axis), due to the harmful chaining, CoQ propagates the error to subsequent iterations and thus shows lower accuracy than UoQ, indicating the vulnerability of CoQ against early errors. In contrast, CoQ has the

[11] In Appendix H and Appendix F, we validate the effectiveness of our verification leveraging answer feedback.

[12] For FaVe in Table 3, due to the insufficient number of training questions, we opted not to fine-tune the answer-aware session generator and instead used the open-source model.

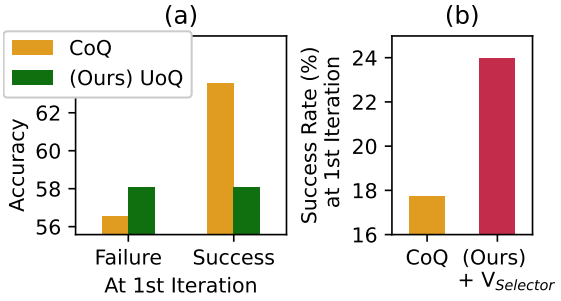

Figure 6: (a) Final accuracy depending on either failure or success in producing gold fact at the first iteration, and (b) improved success rate of CoQ via our unified selector, denoted by "+ $V_{Selector}$".

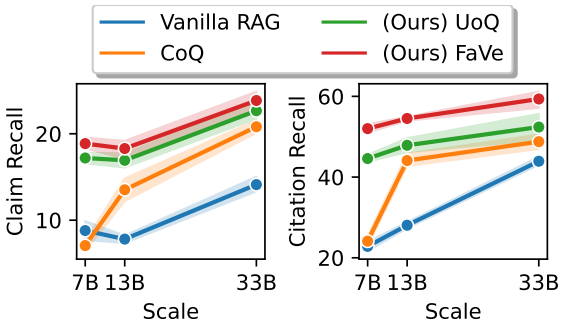

Figure 7: Scaling analysis presenting evaluation results of long-form answers from LLMs with different scales: `vicuna-7B/13B/33B`.

potential to become more accurate if such errors could be avoided, showing hugely improved accuracy given the successful first iteration. However, Figure 6(b) shows that the success rate of CoQ is less than 18%. This suppresses the potential of CoQ, explaining the superior performance of UoQ over CoQ (Table 3).

To realize the potential and enable error-free multi-hop reasoning, by spending additional costs, our unified selector can be easily integrated with CoQ. Specifically, at each iteration of CoQ, we sample multiple SQs, verify them using our unified selector, answer only the verified SQ, and then proceed to the next iteration[13]. For the unified verification feedback r (Eq 2), to avoid the dominance of r^q and the exclusion of r^d due to the binary score of r^q (i.e., either 0 or 1), we set r^q by the probability of "Yes", which is a real number between 0 and 1 as in r^d. Figure 6(b) shows that our unified selector, denoted by "$V_{Selector}$", significantly improves the success rate at the first iteration. Note that, even if the failure occurs, $V_{Selector}$ enables repairment at subsequent iterations, by verifying every reasoning step.

As a result, as shown in Table 3, $V_{Selector}$ significantly improves CoQ performance, realizing its potential. $V_{Selector}$ also shows greater improvements than self-verification (denoted by "w/ self-verification" in Table 3) which ablates the search engine feedback from $V_{Selector}$, indicating the effectiveness of our unified feedback. Nevertheless, it incurs huge latency costs, showing higher overall latency than FaVe by a factor of 3. We suggest adopting our CoQ extension when the best-performing model is needed, while FaVe becomes more suitable when efficiency matters.

Across LLMs with different model sizes and reasoning capabilities, FaVe consistently outperforms both vanilla RAG and CoQ. We further evaluate long-form answers from `vicuna` with varying scales such as 7B, 13B, and 33B. Results are shown in Figure 7.

For the smallest model (7B), CoQ underperforms vanilla RAG. This is because a small model with limited capacity is prone to incur errors and is vulnerable to error propagation, making the limitation of CoQ more pronounced. UoQ and FaVe, breaking the harmful chain, hugely outperform both baselines, indicating that ours are go-to methods, especially for cost-sensitive scenarios.

For the largest model (33B), though hugely improving the claim recall, CoQ shows relatively small improvements in the citation recall compared to vanilla RAG. This is because errors are often attributed to the search failure regardless of LLM's capability, while the search quality significantly influences the citation quality as discussed in Figure 12 and 5. In contrast, by leveraging the search engine feedback, FaVe shows much better citation recall compared to other approaches.

5 Conclusion

In this work, targeting the long-form question-answering task, we aim to improve both the effectiveness (in terms of comprehensiveness and verifiability) and the efficiency (in terms of latency). We propose FaVe, consisting of union-of-query, as a better alternative to chain-of-query, and session verification. Evaluated on ELI5 and StrategyQA, FaVe outperforms baselines with lower latency.

[13]While adapting CoQ, we exclude the dependency between answers, as it degrades the citation quality (Figure 4).

Limitations

Our work produces more accurate citations, enabling the user to verify the generated claims via citation documents. Nevertheless, as an inherent limitation of fact verification, the documents may contain misinformation. For example, when relying on unreliable corpora, e.g., social media posts, the citation documents may include fake news (Webb et al., 2016). As a remedy, for citation sources, we employed trustworthy corpora, i.e., Wikipedia. To further improve reliability and truthfulness, a future work can be incorporating multiple sources for verifiable claims, where citations are cross-checked against multiple sources within the trusted corpus. This helps ensure that the information being cited is not only accurate but also consistent across different references.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (RS-2024-00414981) and ITRC (Information Technology Research Center) support program (IITP-2025-2020-0-01789) supervised by IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

Yihe Deng, Weitong Zhang, Zixiang Chen, and Quanquan Gu. 2023. Rephrase and respond: Let large language models ask better questions for themselves. *arXiv preprint arXiv:2311.04205*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Hangfeng He, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2021. Contrastive learning with adversarial perturbations for conditional text generation. In *International Conference on Learning Representations*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Dmytro Okhonko, Samuel Broscheit, Gautier Izacard, Patrick Lewis, Barlas Oğuz, Edouard Grave, Wen-tau Yih, et al. 2021. The web is your oyster-knowledge-intensive nlp against a very large web corpus. *arXiv preprint arXiv:2112.09924*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring attribution in natural language generation models. *Computational Linguistics*, pages 1–66.

Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94*, pages 232–241. Springer.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Tijmen Tieleman and Geoffrey Hinton. 2012. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn*, 17.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Helena Webb, Marina Jirotka, Bernd Carsten Stahl, William Housley, Adam Edwards, Matthew Williams, Rob Procter, Omer Rana, and Pete Burnap. 2016. Digital wildfires: hyper-connectivity, havoc and a global ethos to govern social media. *ACM SIGCAS Computers and Society*, 45(3):193–201.

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.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575.

Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023. Rewoo: Decoupling reasoning from observations for efficient augmented language models. *arXiv preprint arXiv:2305.18323*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

A Prompt

A.1 CoQ

Figure 8 presents the prompt used for CoQ: It alternates between Prompt A and B, iteratively gener-

Instruction: Write an answer for the given question using only the provided search results and cite them properly by using [1][2][3]. You can use "Search: key words" to retrieve relevant documents and use "Output:" to output an answer. Use "[END]" to end the generation.

(*1-shot demonstration*)
Question: *What's difference between Shia vs. Sunni Islam?*
Answer: *Search: When did the difference first emerge?*
Output: This difference is first formed after ...
Search: ... Output: ... Search: ... Output: ... ... [END]

Question: {{The original question given by a user }}
Answer: ...
Output: {{ The previously generated answer }}
Search: {{ completion: a subquestion (SQ) }}

Prompt A. for subquestion generation

... (using the same instruction & demo to Prompt A) ...
Output: {{ The previously generated answer }}
Search: {{ The generated SQ from Prompt A}}
Document [1]: ... [2]: ... {{ top-2 documents from the SQ }}
Output: {{ completion: answer sentences to the SQ }}

Prompt B. for answer generation

Figure 8: CoQ alternates between Prompt A/B.

Instruction: Generate a search session consisting of detailed queries relevant to the question.

(*1-shot demonstration*)
Question: *difference between Shia vs. Sunni Islam*
Search Session:
[1] When did the difference first emerge?
[2] What is the difference in the ideological practice?
[3] How do the practices and rituals differ?

Question: {{ given question from the user}}
Search Session:
{{ completion: a sequence of subquestions}}
[1] (1st subquestion) ...
...
[N] (N-th subquestion) ... [END of subquestions]

Figure 9: Prompt for search session generation.

(8-shot demonstrations)
Question: *What's the difference between Shia vs. Sunni Islam?*
Sub-topic: *What is the difference in the ideological practice?*
Will the sub-topic answer the question? *Yes*
...
Question: *Why can't the US just copy healthcare systems such as the UK and Canada?*
Sub-topic: *How does Canada's healthcare system differ from the UK system?*
Will the sub-topic answer the question? *No*

Question: {{ given question from the user}}
Sub-topic: {{ generated sub-question }}
Will the sub-topic answer the question? {{ completion:Yes/No }}

Figure 10: Prompt used for the internal feedback of relevance for SQ to the given question

ating SQs and providing answers, conditioned by preceding responses (highlighted in red).

(Ours) FaVe vs baseline	FaVe Wins	Tie	FaVe Loses
vs Vanilla RAG	**58.7%**	15.3%	26.0%
vs CoQ	**45.0%**	15.4%	39.6%

Table 4: Human evaluation results on reference-free pairwise comparisons.

A.2 UoQ

Figure 9 shows the prompt for the search session generation, producing a sequence of N numbers of SQs.

Figure 10 presents the prompt to collect the internal feedback for verifying the generated SQs. If $\tilde{q}_i$ is classified as relevant (i.e., "Yes" in the prompt), r_i^q is set by 1 and, otherwise 0.

For the answer generation, we used the same prompt to CoQ (Prompt B in Figure 8), while excluding previous answers and SQs from generating answers.

B Human evaluation

FaVe is preferred more often than baselines by human annotators. In addition to results from automatic measures, we present human evaluation results on ELI5 dataset.

Though enabling sufficient scale evaluation, the automatic metrics suffer from missing annotations on relevant facts. For ELI5, for example, given that only a single reference gold answer is available for each question, we found that 60-80% of gold claims in model-generated answers are not captured by the reference answer. As a reference-free alternative, three human annotators were asked to directly compare each pair of models regarding the answer comprehensiveness, on randomly subsampled 50 questions. We used Amazon Mechanical Turk, and each annotator was paid 0.1$ for each assignment.

Table 4 reports human evaluation results of reference-free pairwise comparisons on ELI5 dataset, showing that FaVe is preferred more often than the others.

C Latency at different stages

Figure 11 compares latency for different RAG approaches.

Reducing latency through factored generation Through factored generation, UoQ improves not only the answer quality but also efficiency, as demonstrated in Figure 11.

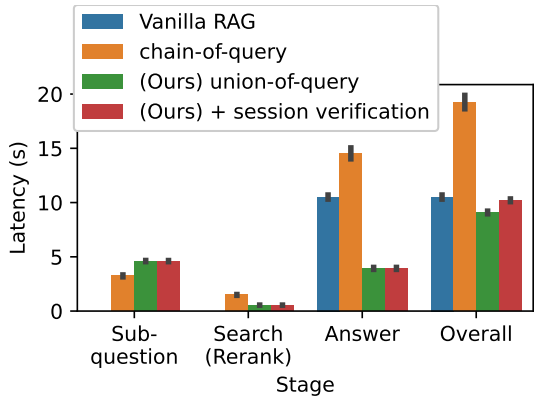

Figure 11: Latency comparison between different RAG approaches at different stages.

Model (vicuna-13b)	Claim Recall	Citation Recall	EM	F1
Vanilla RAG	11.8	33.1	22.5	32.1
CoQ	10.6	48.5	26.7	35.6
(Ours) UoQ	11.0	57.6	27.6	36.8
(Ours) FaVe	**19.6**	**65.2**	**28.4**	**38.7**

Table 5: Results on HotpotQA dataset. EM denotes exact-match scores between the final prediction and the ground-truth answer.

CoQ shows the worst overall latency, as it sequentially calls LLM multiple times with increasingly lengthy prompts (Xu et al., 2023). In contrast, UoQ maintains comparable overall latency to vanilla RAG, by avoiding sequential answer generation. The subquestion generation is the only part where the sequential process is involved, yet incurs relatively lower latency, as it takes shorter inputs and outputs (i.e., the given questions and SQs, respectively) compared to answer generation which takes long documents.

Session verification with little latency costs The session verification incurs little latency overhead, as illustrated in Figure 11 with "+ session verification". The only additional latency costs arise from employing the internal feedback to compute r_i^q, which remain negligible as it takes short SQs. The two external feedbacks incur no additional costs, since we employ the answer feedback only for training and, for the search engine feedback, reuse the document relevance scores.

D Results on HotpotQA

Table 5 compares performance on HotpotQA dataset (Yang et al., 2018). For the evaluation metrics, we report the claim recall (using annotated supporting facts provided in the dataset, as gold claims) and citation recall, along with the official metrics of the dataset for the final answer accuracy, i.e., Exact-Match and F1 on the final answer string.

Model (vicuna-13b)	Claim Recall	Citation Recall	EM	F1
UoQ	11.0	57.6	27.6	36.8
+ $V_{Generator}$	**16.7**	**58.7**	**28.5**	**37.3**

Table 6: Showcasing the transferability of our answer-aware session generator, denoted by $V_{Generator}$, fine-tuned using ELI5 and then evaluated on HotpotQA.

Consistent with results on ELI5 (Table 1) and StrategyQA (Table 3), results on HotpotQA demonstrate that UoQ outperforms CoQ, while FaVe further achieves significant improvements, surpassing all baselines across all metrics. These results validate the generalizability of our method across diverse datasets.

We further leverage HotpotQA dataset, to examine the transferability of our answer-aware session generator. Specifically, we first fine-tune the search session generator using ELI5 dataset, and then transfer it to HotpotQA. Results are reported in Table 6, where "+ $V_{Generator}$" denotes replacing the open-source `vicuna-13b` model for search session generator in UoQ by our answer-aware generator fine-tuned using ELI5. The answer-aware session generator improves performance on all metrics, showcasing the transferability of the fine-tuned model using sufficiently diverse questions in ELI5 dataset.

E Results from different search methods

UoQ, improving queries, is complementary to employing a better retriever, to facilitate retrieval and RAG. Orthogonal to UoQ improving search queries via better SQs, one can employ a more effective retriever, to enhance RAG via improved search results. In particular, we compare three retrievers: BM25 (Robertson and Walker, 1994), which relies on lexical-exact-match and is the least accurate retriever; dense retriever (Ni et al., 2022)[14], demonstrating moderate accuracy; and cross-encoder (Nogueira et al., 2020)[15], considered

[14] https://huggingface.co/sentence-transformers/gtr-t5-large

[15] https://huggingface.co/castorini/monot5-base-msmarco

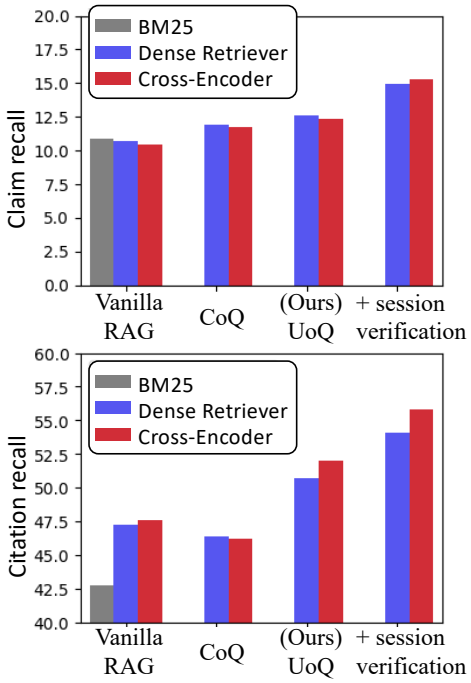

Figure 12: Performance from different retrievers.

the most accurate among them. We report performance on two task objectives in Figure 12.

Compared to BM25, employing better retrievers (i.e., the dense retriever or the cross-encoder) significantly enhances citation recall, as illustrated by vanilla RAG in the lower figure. Nevertheless, there are little differences in claim recall, as shown in the upper figure. In contrast, ours, improving search queries, surpasses all baselines in both claim recall and citation recall, regardless of the retriever choice. This underscores the unique contribution of leveraging better SQs via the chain break and session verification.

Furthermore, regarding the citation recall, UoQ exhibits a synergistic advantage when paired with the most accurate retriever, cross-encoder: Compared to the dense retriever, when the cross-encoder is employed, the citation recall of UoQ increases by 1.7%pt. In contrast, such benefits are not observed in vanilla RAG and CoQ. This is because improved search results can be achieved only when a high-quality query is used in conjunction with a high-performing retriever.

F Results of using test-time answer feedback

Leveraging the answer feedback for assessing both comprehensiveness and verifiability achieves the best performance on both objectives. To complement our answer-aware session generator, which assesses session comprehensiveness using answer feedback at *train time*, the answer feedback can be further leveraged at *test time* to assess the verifiability of answer samples, albeit with additional latency costs. Table 7 presents the contribution of leveraging the answer feedback (AF) at train or test time, denoted by "train-time AF" and "test-time AF", respectively.

Model	LLM	Claim Recall	Citation Recall
Vanilla RAG	ChatGPT	12.0	51.1
+ test-time AF		11.4	69.3
(Ours) FaVe	vicuna-13b	**15.3**	55.8
+ test-time AF		<u>15.0</u>	**74.1**
w/o train-time AF		12.9	<u>71.0</u>

Table 7: "AF" denotes the answer feedback given at either train time or test time. The **bold**-faced and the <u>underlined</u> denote the best and the second best performance, respectively.

Specifically, for test-time answer feedback, we sample multiple answers[16], evaluate each on the citation recall, and select the best answer as the final response. Note that, in contrast to the claim recall, the citation recall can be measured without the need for gold reference answers. We compare FaVe (employing `vicuna-13b`) with or without test-time answer feedback, along with vanilla RAG that uses ChatGPT as a baseline.

Leveraging test-time answer feedback improves citation recall. Nevertheless, as expected, it does not improve claim recall, highlighting the unique contribution of FaVe to answer comprehensiveness. As a result, FaVe leveraging both train- and test-time answer feedback outperforms ChatGPT baselines on both metrics. Meanwhile, removing train-time answer feedback decreases performance, indicating the complementarity between the two answer feedbacks.

G Results on Knowledge Utilization

UoQ improves and better utilizes search results. As an attribution of improved task performance, we demonstrate that UoQ contributes to enhancing RAG by better acquiring and utilizing knowledge through retrieved documents.

To this end, we introduce a metric, termed "effective knowledge coverage", to evaluate how much gold knowledge is covered in retrieved documents and effectively utilized in the final answer. As

[16]We sampled 8 answers per question.

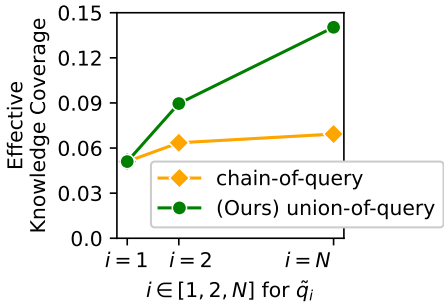

Figure 13: Effective knowledge coverage at different search iterations.

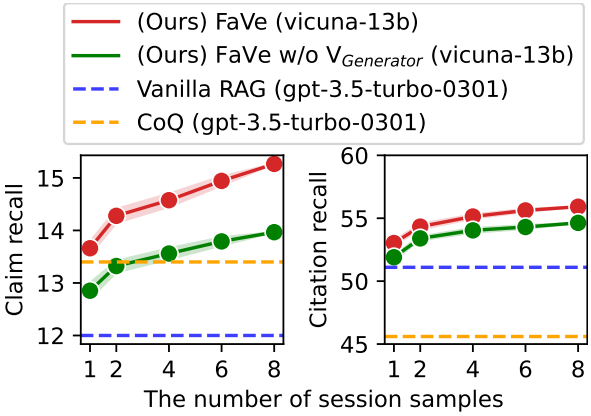

Figure 14: Comparison between FaVe and FaVe without the verification via the answer-aware session generator (denoted by "FaVe w/o $V_{Generator}$"), on sample efficiency of session candidates.

gold knowledge, we used gold factual claims open-sourced by Gao et al. (2023b). We denote the set of gold factual claims by $\mathcal{C}$, and those included in the retrieved documents and the model-generated answer at different iterations by $\mathcal{C}_d^i \subset \mathcal{C}$ and $\mathcal{C}_a^i \subset \mathcal{C}$, respectively, where i denotes the index of the search query $\tilde{q}_i$. To obtain $\mathcal{C}_d^i$ and $\mathcal{C}_a^i$, we examine whether each gold claim $c \in \mathcal{C}$ is entailed from the documents and answers, respectively, by employing the same NLI model used for the claim recall and the citation recall. Finally, the effective knowledge coverage is computed by $|\bigcup_i \mathcal{C}_d^i \cap \mathcal{C}_a^i|/|\mathcal{C}|$, where $|\cdot|$ denotes the cardinality of a set. Note that effective knowledge coverage differs from claim recall, as it assesses the knowledge in the documents in conjunction with the answer. Results are reported in Figure 13.

CoQ (orange curve) shows marginal improvement across iterations, indicating that the harmful chaining prevents the model from acquiring and utilizing comprehensive gold knowledge. In contrast, UoQ (green curve), breaking the chain, consistently improves the coverage.

H Results on sample efficiency

Answer feedback enables the session candidate generator to produce high-quality candidates, improving the sample efficiency. In Figure 14, we showcase the effectiveness of the answer-aware session generator trained using the external feedback on answers, in terms of the sample efficiency of search session candidates. Specifically, we compare FaVe and FaVe without verification via the answer-aware session generator (denoted by "FaVe w/o $V_{Generator}$"), with varying numbers of session candidates (among which the best session is selected). In addition, as strong baselines, we compare the performance of vanilla RAG and CoQ that use ChatGPT as backbone LLM.

FaVe consistently shows better performance than FaVe w/o $V_{Generator}$ on both metrics, improving the overall sample efficiency. Meanwhile, for both FaVe and FaVe w/o $V_{Generator}$, the performance consistently improves as we increase the number of session candidates, showing the effectiveness of the unified session selector and complementarity between the two.

I Use Or Create Scientific Artifacts

We used evaluation metrics proposed by Gao et al. (2023b), under MIT license, for research purpose on. It covers open-domain questions and answers, in English.

J Potential Risk

Since we mainly target improving answer comprehensiveness and verifiability, our model may produce harmful or offensive responses.

K Usage of AI Assistant

We used ChatGPT for language edits.