ECoRAG: Evidentiality-guided Compression for Long Context RAG

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have shown remarkable performance in Open-Domain Question Answering (ODQA) by leveraging external documents through Retrieval-Augmented Generation (RAG). To reduce RAG overhead, from longer context, context compression is necessary. However, previous compression methods do not focus on filtering out non-evidential information, which limit the performance in LLM-based RAG. We thus propose Evidentiality-guided RAG, or **ECoRAG** framework. ECoRAG improves LLM performance by compressing retrieved documents with a focus on evidentiality, ensuring whether answer generation is supported by the correct evidence. As additional step, ECoRAG reflects whether the compressed content provides sufficient evidence, and if not, retrieve more until sufficient. Experiments show that ECoRAG improves LLM performance on ODQA tasks, outperforming existing compression methods. Furthermore, ECoRAG is highly cost-efficient, as it not only reduces the total RAG latency but also minimizes token usage by retaining only the necessary information to generate the correct answer. The code is publicly available for further exploration.[1]

1 Introduction

LLMs (OpenAI, 2023; Touvron et al., 2023) have excelled in tasks such as ODQA by leveraging external knowledge through RAG (Lewis et al., 2020; Ram et al., 2023). However, RAG inevitably grows context length, which incurs computational cost and also hinders generation quality (Liu et al., 2024; Hsieh et al., 2024; Li et al., 2024).

While adopting existing context compression (Li et al., 2023) may look promising, such a baseline presents two main challenges. First, LLMs are known to be vulnerable to irrelevant contents that cannot provide evidence for generated content (Shi et al., 2023; Qian et al., 2024; Wu et al., 2024), and existing compression methods (Xu et al., 2024; Jiang et al., 2024; Yoon et al., 2024) fail to sufficiently filter them out. As a result, a naive baseline simply prepending retrieved documents, 'Raw Prepend' in Figure 1, outperforms a baseline compressor RECOMP (Xu et al., 2024). As the number of documents increases, a baseline compressor fails to filter out increasing irrelevant contents, causing performance to decline.

Second, it is challenging to determine the desirable compression ratio for each question. Failure to do so may lead to compressing too much, which results in losing crucial information, or compressing too little, which produces overly long contexts that degrade generation quality (Liu et al., 2024;

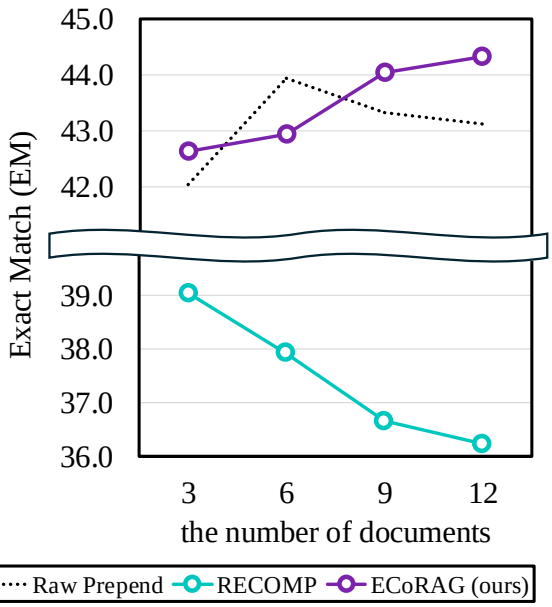


Figure 1: Comparison of performance between prepending retrieved documents (Raw Prepend) (Karpukhin et al., 2020), applying RECOMP (Xu et al., 2024), and applying ECoRAG on the Natural Questions (Kwiatkowski et al., 2019) test set. Experiments were conducted using Flan-UL2 (Tay et al., 2023).

[1] https://anonymous.4open.science/r/ecorag-54BF

Hsieh et al., 2024; Li et al., 2024) and increase computational costs. Thus, it is necessary to find the desirable compression ratio that can generate the correct answer for each question.

Our distinction is using evidentiality to address both challenges and proposing **E**videntiality-guided **Co**mpression and **R**etrieval-**A**ugmented **G**eneration (**ECoRAG**) framework: Ours compresses retrieved documents to retain only the information necessary to support the answer. To overcome the first challenge, evidentiality (Lee et al., 2021; Asai et al., 2022) is used to determine whether each sentence in the retrieved documents supports the correct answer to a question. It can be quantified for each sentence by measuring how much it contributes to the model to generate the correct answer. We train the compressor using this as training signals.

To address the second challenge, ECoRAG reflects on compression as a collective, where it contains sufficient evidence. We begin by forming the smallest possible collective unit of compression and assess whether it is evidential. If not, it means that it is compressed too much, which we adjust adaptively by collecting more, until it is sufficient. Through this reflection process, ECoRAG finds the desirable compression ratio that enables the LLM to generate the correct answer with minimal tokens.

By applying these methods, ECoRAG has two advantages when dealing with long contexts as the number of documents increases. First, ECoRAG improves performance by retaining only the information necessary for generating the correct answer and removing distracting content. This results in gains on ODQA datasets such as Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (TQA) (Joshi et al., 2017). Second, by compressing the long context to only what is needed, it reduces computational costs.

Our contributions to this work can be summarized as follows: (1) Evidentiality-guided Compression: We developed a method that compresses retrieved documents based on evidentiality. (2) Evidentiality Reflection for Adaptive Compression: Our framework evaluates compressed content for evidentiality and adaptively adjusts the length of compression. (3) Experimental results show that our approach significantly improves retrieval-augmented LLM performance on ODQA datasets. (4) Our approach is also cost-efficient, as it quickly compresses long context, reducing latency and tokens.

2 Related Work

2.1 Evidentiality-guided RAG

Dense retrievers (Karpukhin et al., 2020; Izacard et al., 2022) focus on lexical answerability, but may mislabel documents as relevant when they lack contextual evidence, leading to the need for evidentailtiy. In prior work (Lee et al., 2021), evidentiality refers to whether a document supports generating the correct answer to a question. Unlike answerability, evidentiality is more challenging to mine directly as it reflects the contextual relationship between a question and a document. To measure evidentiality, previous work checks whether the removal of the document is critical for answering the question (Asai et al., 2022), utilizes attention scores (Niu et al., 2020), or considers the change in confidence scores (Song et al., 2024). Our work introduces evidentiality in LLMs, enhancing RAG by prioritizing contextually rich documents for generating correct answers.

2.2 Prompt Compression

Numerous studies (Mu et al., 2024; Li et al., 2023; Kim et al., 2024) have focused on prompt compression to address both cost and performance challenges, as shown in prior research (Shi et al., 2023; Liu et al., 2024; Hsieh et al., 2024). RECOMP (Xu et al., 2024) provides both extractive and generative summaries of documents, considering whether the summaries helped answer the given question. LLMLingua (Jiang et al., 2023) uses conditional probabilities of LLMs to guide fine-grained prompt compression. Building on this, LongLLMLingua (Jiang et al., 2024) compresses prompts in long context scenarios by using a question-aware coarse-to-fine compression and document reordering mechanism. Similarly, CompAct (Yoon et al., 2024) employs an adaptive compression strategy to iteratively compress documents while retaining key information relevant to the query. However, existing methods struggle to compress long context, which prevents them from fully utilizing the retrieval results.

2.3 Retrieval Evaluation for RAG

LLMs may evaluate the quality of retrived results for enhancing RAG, as seen in Madaan et al. (2024), where models iteratively improve their responses; this concept has been applied to RAG. Self-RAG (Asai et al., 2024) trains LLM to evaluate retrieved documents and its output by predicting reflection tokens that assess the need for retrieval

and the quality of the generated text. Labruna et al. (2024) dynamically determines whether to retrieve additional context when needed by using a trained reader LLM. CRAG (Yan et al., 2024) employs a retrieval evaluator to assess document relevance and triggers corrective actions to refine retrieved information, by using lexical overlap between questions and documents. In our ECoRAG framework, we evaluate whether the evidence is sufficient to generate the correct answer by leveraging evidentiality as defined by the LLM.

3 Proposed Method

In this section, we describe how ECoRAG adaptively adjusts the compression length to ensure that the LLM generates the correct answer. To achieve this, we focus on: (1) compressing retrieved documents by sorting them based on evidentiality (Section 3.1), and (2) evaluating whether the compressed document is sufficiently evidential, and if not, adaptively incorporating more information (Section 3.2), and Figure 2 provides an overview.

3.1 Evidentiality-guided Compressor

This section explains how the retrieved documents are compressed to preserve the evidence that enables the LLM to generate the correct answer. To retain the necessary content and remove irrelevant parts during the compression process, we extract evidential sentences from the retrieved documents (Section 3.1.1) and use them to train the compressor (Section 3.1.2).

3.1.1 Definition of Evidentiality

We define the evidentiality of a sentence by prioritizing how much it contributes to generating the correct answer, while penalizing distractors that interfere with this process. The degree of evidentiality is categorized hierarchically based on two conditions. We find sentences that enable the LLM to generate the correct answer. If a sentence does not, then we check if it interferes with other evidence.

First, when assessing whether each sentence helps generate the correct answer, it is important to consider that the LLM contains parametric knowledge (Wang et al., 2020; Yu et al., 2023; Luo et al., 2023). Prior work (Lee et al., 2021; Asai et al., 2022) has focused on whether the language model could contribute to generating the correct answer using given document. However, it is challenging to distinguish whether the correct answer was generated using the document or parametric knowledge, especially in larger models. If the correct answer was generated solely using parametric knowledge, regardless of the given document, it is difficult to determine whether the document serves as key evidence. Therefore, we propose the following first condition: **1** Without the sentence the LLM cannot generate the correct answer alone, but with the sentence it can.

Second, it is also crucial for the compressor to filter out distractors that hinder the evidence from generating the correct answer. While it is possible to train robustness against distractors through instruction fine-tuning (Liu et al., 2024), LLMs often require substantial costs for training and closed LLMs often impossible to train. If the compressor can remove distractors, it can be applied to any LLM without requiring additional training. To identify distractors, we introduce a second condition for sentences that do not meet **1**: **2** The sentence does not interfere with the evidence defined in **1** in generating the correct answer.

Based on the aforementioned conditions, we hierarchically define evidentiality as depicted in Figure 3. Sentences satisfying condition **1** are labeled as **strong evidence**. Sentences failing to meet condition **1** are further classified based on condition **2**: those meeting condition **2** are labeled as **weak evidence**, while those that do not are classified as **distractor**.

3.1.2 Learning Objective for Compressor

Given sentences $\mathcal{D} = \{d_1, d_2, \cdots, d_{|\mathcal{D}|}\}$, for a question q, we train our compressor based on dual encoders (Izacard et al., 2022) to differentiate between strong and weak evidence, as well as distractor. Using dual encoders, E_Q for questions and E_D for sentences, we calculate the similarity score between q and sentences in $\mathcal{D}$ (i.e., $sim(q, d_i) = E_Q(q) \cdot E_D(d_i)$). Sentences are categorized into strong (d^*) or weak (d^+) evidence, and distractor (d^-) based on our hierarchical definition. We define similarity scores as $s^* = sim(q, d^*)$, $s^+ = sim(q, d^+)$, and $s^- = sim(q, d^-)$. The similarity scores are utilized to train two inequalities:

$$(s^+ > s^-), \quad (s^* > s^+, s^-) \tag{1}$$

These inequalities ensure that strong evidence is ranked above weak evidence, which in turn is ranked above distractor, guiding the training of our compressor.

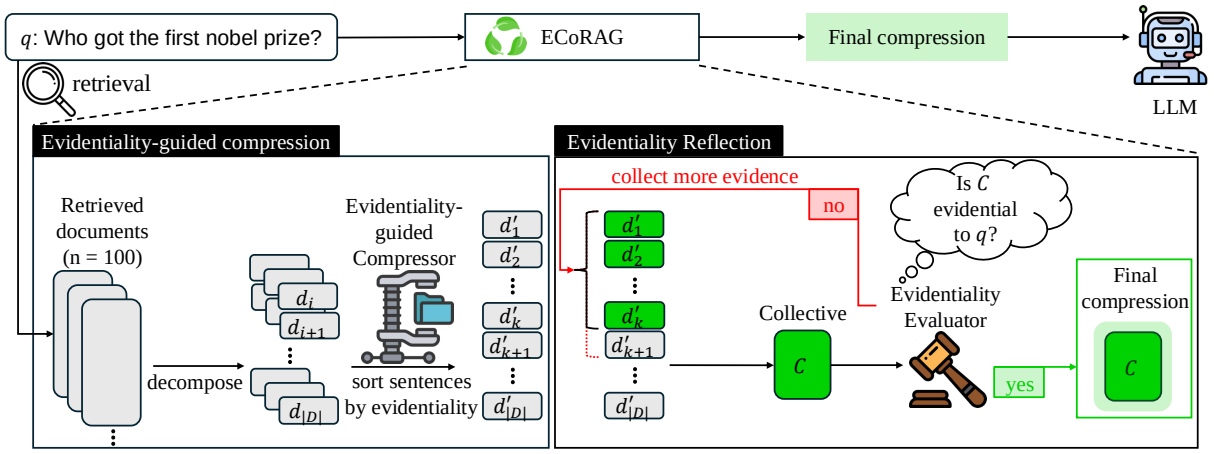


Figure 2: This figure illustrates the overall framework of ECoRAG. First, the evidentiality-guided compressor compresses the retrieved documents by sorting decomposed sentences based on evidentiality. In evidentiality reflection, our evaluator assesses whether the compressed document C is evidential as a collective of sentences. If the compressed document is evidential, it is used for final compression (green line). If not, the compression rate is adjusted to add more evidence (red line).

The weak evidentiality loss $\mathcal{L}_{we}$ uses the InfoNCE loss to distinguish weak evidence d^+ from distractor d^-. The loss function is formulated as:

$$\mathcal{L}_{we} = -\log \frac{\exp(s^+/\tau)}{\exp(s^+/\tau) + \sum_{d_j^- \in D^-} \exp(s_j^-/\tau)} \quad (2)$$

Here, $s_j^- = sim(q, d_j^-)$ represents the similarity score for each distractor in the set D^-, and τ is a temperature parameter.

The strong evidentiality loss $\mathcal{L}_{se}$ also utilizes the InfoNCE loss to prioritize strong evidence d^*. The loss function is formulated as:

$$\mathcal{L}_{se} = -\log \frac{\exp(s^*/\tau)}{\exp(s^*/\tau) + \sum_{d_j^\pm \in D^- \cup D^+} \exp(s_j^\pm/\tau)} \quad (3)$$

Here, $s_j^\pm = sim(q, d_j^-)$ is the similarity score for each sentence in the combined sets of distractors D^- and weak evidences D^+.

The final loss $\mathcal{L}$ is defined as the sum of the strong evidentiality loss and the weak evidentiality loss:

$$\mathcal{L} = \mathcal{L}_{se} + \mathcal{L}_{we} \quad (4)$$

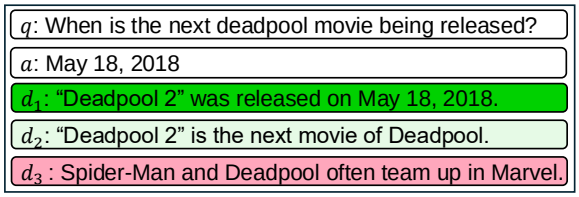
Our compressor is trained using this loss $\mathcal{L}$, and ranks sentences $d'_1, d'_2, \ldots, d'_{|\mathcal{D}|}$ by evidentiality, selecting high-scoring ones for compression. The number of sorted evidence required can vary depending on the difficulty of each question. However, providing too little evidence may omit important information, while too much increases computational costs for each question. Thus, balanced compression ratio is necessary for each question to address both issues.

3.2 Evidentiality Reflection for Adaptive Compression

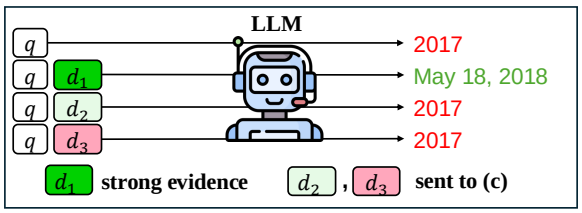
Once a collective of evidential sentences is collected, we need to determine whether compression ratio is desirable. For this task, we reflect on the given compression, using language model to reflect on the evidentiality of compressed document (Section 3.2.1). Then, if compressed too much, we adaptively adjust the compression ratio by collecting more (Section 3.2.2).

3.2.1 Training Evidentiality Evaluator

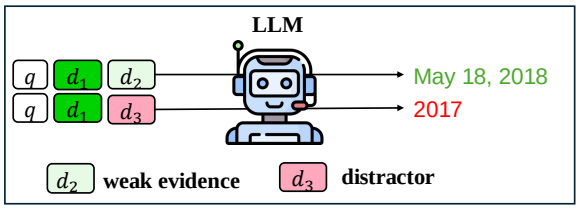
We construct effective **evidentiality evaluator** $\mathcal{M}_{eval}$ that assesses whether the compressed document is strong evidence enough to generate the correct answer. In prior work, CompAct (Yoon et al., 2024) trained the evaluator by prompting GPT-4o (OpenAI, 2023) to determine if the evidence is sufficient to answer the question. However, this approach can introduce bias (Chiang and Lee, 2023) when GPT-4o evaluates through prompting, leading to inaccurate supervision. Accurate supervision requires verifying if the document actually enables the reader LLM to generate the correct answer. Therefore, we reuse our evidentiality labels obtained from the LLM in Section 3.1.1 and distill them from our reader LLM into smaller model, Flan-T5-large (Chung et al., 2022), to build the evaluator. The performance comparison between CompAct and our evaluator is discussed in Section



(a) Example of question, answer, and sentences for evidentiality mining



(b) strong evidentiality mining



(c) weak evidentiality mining

Figure 3: This figure illustrates the evidentiality mining strategy of ECoRAG.

5.2.

We utilize our evidentiality labeled dataset (d^*, d^+, d^-) for training the evidentiality evaluator $\mathcal{M}_{eval}$ to determine if compressed document is sufficient for correct answer generation. The evidentiality evaluator is trained to distinguish whether the given compressed document is strong evidence to enable the LLM to generate the correct answer. Thus, we add 2 special tokens $t \in$ [<EVI>, <NOT>] and train $\mathcal{M}_{eval}$ to generate '<EVI>' for strong evidence d^*, and '<NOT>' for other sentences d^+, d^-. Subsequently, next-token prediction loss $\mathcal{L}_{eval}$ is used for this training stage to predict whether compressed document is strong evidence.

$$\mathcal{L}_{eval} = -log\ p_{\mathcal{M}_{eval}}(t|q, d) \quad (5)$$

3.2.2 Adaptive Compression

In adaptive compression, the compression ratio is adaptively adjusted by our evaluator, which reflects on whether the current compression is evidential, as described in Figure 2. Initially, our evaluator assesses the evidentiality of compressed document C containing only the first evidence, d'_1, from our ordered evidences $d'_1, d'_2, \ldots, d'_{|\mathcal{D}|}$. If the evaluator determines that C is evidential, it becomes the final compression provided to LLM. If C is not evidential, we add the next piece of evidence, d'_2 is added to d'_1 to build a new compressed document; when the k-th iteration fails, d'_{k+1} is added to the previous compressed document. This process is repeated until the desirable compression is found, with a token limit set to avoid infinite loop. The final compression is then used as input for the LLM, which generates the final answer.

Although iterative adjustment can increase latency compared to using raw documents, ECoRAG reduces it efficiently. Prior work (Yoon et al., 2024), each iteration required LLM (7B) to generate a new compression by using the previous compression and the next piece of evidence. Thus, with each iteration, LLM reads different contents and generates compression of multiple tokens, increasing latency time. However, ECoRAG reduces redundancy by ordering evidence just once and adding it iteratively. Moreover, our framework utilized a lightweight evaluator (0.77B) that adjusts compression length by generating just a single special token, resulting in rapid compression speed; the actual results are shown in Section 5.4.

4 Experiments

4.1 Experimental Settings

Datasets We evaluate our framework through NQ (Kwiatkowski et al., 2019), TQA (Joshi et al., 2017), and WQ (Berant et al., 2013), which are ODQA benchmark datasets. We use the top 100 documents retrieved from DPR (Karpukhin et al., 2020)[2].

Base Language Models We initialize our evidentiality compressor from Contriever (Izacard et al., 2022) checkpoint pre-trained on CC-net (Wenzek et al., 2020) and English Wikipedia (Izacard et al., 2022). We use Contriever to compare its performance with RECOMP (Xu et al., 2024). For evidentiality evaluator, we utilize Flan-T5-large (Chung et al., 2022). For the reader model, we use GPT-4o-mini (OpenAI, 2023)[3], as it supports a context length of 128K tokens, sufficient to process all 100 retrieved documents.

[2]Since enhancing the retriever is beyond the scope of this study, we conduct our experiments under the assumption that the retrieved documents are already provided.

[3]gpt-4o-mini-2024-07-18

Methods	NQ			TQA			WQ		
	#tokens ↓	EM	F1	#tokens ↓	EM	F1	#tokens ↓	EM	F1
RAG without compression									
closed book	0	31.88	44.10	0	64.78	73.10	0	24.51	42.73
100 documents (Karpukhin et al., 2020)	13905	36.09	**50.18**	14167	56.21	64.22	13731	21.11	38.72
RAG with 100 documents compressed									
LLMLingua (Jiang et al., 2023)	635	26.84	38.30	630	50.81	57.91	641	22.98	39.77
LLMLingua-2 (Pan et al., 2024)	1315	30.11	42.52	1324	53.19	60.46	1113	23.52	40.61
LongLLMLingua (Jiang et al., 2024)	1370	32.96	45.32	1402	55.75	63.75	1355	21.51	39.13
RECOMP (extractive) (Xu et al., 2024)	662	32.85	44.54	672	51.66	59.08	658	19.54	36.83
RECOMP (abstractive) (Xu et al., 2024)	**14**	27.59	39.19	**26**	39.95	46.68	**19**	20.47	36.90
CompAct (Yoon et al., 2024)	106	35.71	47.14	96	63.96	73.87	75	29.77	44.25
ECoRAG (ours)	632	**36.48**	49.81	441	**65.34**	**75.37**	560	**30.17**	**46.13**

Table 1: Compression methods performance comparison on NQ, TQA, and WQ. The table shows the results using GPT-4o-mini as the reader model, given 100 retrieved documents. It reports the number of tokens after compression, along with EM and F1-score, illustrating the impact of different compression methods on model performance.

Evaluation Metrics We report results on the test sets of NQ, TQA, and WQ using EM and word-level F1-score to assess the question-answering task performance. We also report the average number of input tokens given to the reader LLM to evaluate the effectiveness of our compression step.

Baseline We report two types of baselines, *RAG without compression* and *RAG with 100 compressed documents*.

RAG without compression: As a baseline, we report the results using only the question and raw retrieved documents. The 'closed book' setting, where no retrieval is used, shows that the model relies solely on its internal knowledge. In the '100 documents' setting, we simply concatenate the top 100 retrieved documents without any compression for evaluation. This is the approach used in conventional RAG without compression.

RAG with 100 compressed documents: We also reproduce several retrieval augmentation methods for comparison. To better understand the effect of different compression methods, we evaluated several baselines including LLMLingua (Jiang et al., 2023), LLMLingua-2 (Pan et al., 2024), LongLLMLingua (Jiang et al., 2024), CompAct (Yoon et al., 2024), and RECOMP which offers both extractive and abstractive variants.

4.2 Results

In this section, we report the results of our model and compare them with other compression baselines for ODQA in Table 1. Other compression methods outperform the closed book setting, where only internal knowledge is used, except for LLMLingua. Furthemore, as shown, our model outperforms other baselines, including the 100-documents setting. This is notable as all other compression methods perform worse than simply prepending the retrieved documents indiscriminately.

In the long context setting, retrieving many documents often brings in those with low relevance scores, introducing noise. As a result, previous compression methods struggle to effectively filter out noise. Notably, ECoRAG outperforms all these methods, even with fewer tokens than some of them. The strength of ECoRAG lies in compressing only the necessary content, focusing solely on the information essential for generating the correct answer, therefore outperforms the strongest compression baseline in NQ (+0.77%p), TQA (+1.38%p), and WQ (+0.40%p) in EM.

From a token efficiency perspective, ECoRAG uses more tokens than RECOMP (abstractive) and CompAct but still outperforms them, while compressing with fewer tokens than other methods. According to Xu et al. (2024), abstractive RECOMP performs well in the 5-document setting, but its ability to effectively compress information diminishes in long context setting due to the limitations on input size. CompAct suffers from inaccurate compression evaluation, failing to add needed information or introducing noise. As a result, it fails to capture the required information with fewer tokens, leading to lower performance. In contrast, ECoRAG can handle long context, effectively retaining only necessary content needed to generate the correct answer, which results in superior performance across different datasets. This demonstrates the strength of ECoRAG in balancing token usage with answer accuracy, making it the most effective approach for long context compression.

Methods	NDCG@1	NDCG@10
Answerability (baseline)	67.82	79.20
Leave-One-Out (Asai et al., 2022)	70.67	80.80
ECoRAG (ours)	**75.53**	**81.92**

Table 2: Comparison of NDCG@1 and NDCG@10 on HotpotQA dataset using different training signals

5 Analysis

In addition to the main results, we verified the effectiveness of our framework by addressing the following research questions:

- **RQ1**: Does our compressor effectively capture human-annotated evidence?
- **RQ2**: How accurately does our evaluator predict evidentiality?
- **RQ3**: What is the impact of each component in ECoRAG?
- **RQ4**: Is ECoRAG efficient compression?

5.1 RQ1: Alignment with Human-annotated Evidentiality

In this section, we verify whether our compressor can effectively sort sentences by evidentiality and provide them to our evaluator. Although our compressor contributes to LLM performance by learning LLM-defined evidentiality, it should be verified whether it effectively captures ground-truth evidence. Thus, we conducted experiment using HotpotQA (Yang et al., 2018), which provides human-annotated evidence. In this experiment, we compared how well prior work and our compressor assign higher scores ground-truth evidence. We use Normalized Discounted Cumulative Gain (NDCG) as a metric to evaluate how effectively evidentiality-focused methods, including ours, rank evidence higher.

As shown in Table 2, ECoRAG achieved the highest performance, aligning well with human-annotated evidentiality. The 'Answerability' means training the compressor by treating passages containing the correct answer as positive and those without as negative. The 'Leave-One-Out' (Asai et al., 2022) labels a passage as positive if removing it prevents the model from generating the correct answer, and negative if the model still succeeds. ECoRAG shows better performance than the prior evidentiality baseline in NDCG@1 (+4.86%p) and NDCG@10 (+1.12%p), indicating that it aligns well with human-annotated evidence and effectively captures evidence. Therefore, our compressor provides well-sorted evidences to our evaluator, then we need to verify the evaluator, the other component of ECoRAG.

5.2 RQ2: Evaluator Performance on Evidentiality Prediction

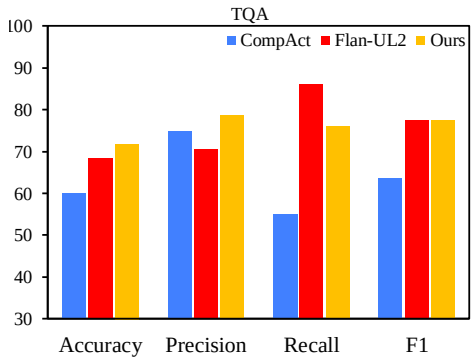


Figure 4: Evidentiality evaluation metrics using different evaluator, including ours, measured on the TQA.

ECoRAG also requires the evidentiality evaluator to accurately assess whether the compressed document can generate the correct answer, and we conducted experiments to verify its accuracy. For each question in the test set of TQA, we define ground-truth labels for retrieved documents as either <EVI>, which leads to generating the correct answer as in Section 3.2.1, or <NOT>. Then, we evaluate how well our evaluator and other evaluators that assess whether each document is necessary for the LLM can predict these labels by measuring accuracy, precision, recall, and F1-score. We report the results of these predictions in Figure 4.

Across all metrics, our evidentiality evaluator effectively predicts evidentiality, even though it is a smaller model than other evaluators. Our evidentiality evaluator outperforms (+13.96%p in F1 scores) the evaluator (7B) from CompAct (Yoon et al., 2024), which is trained using supervision from GPT-4o to evaluate its compressed document based on provided information without considering additional background context. According to Asai et al. (2024), the reader LLM evaluates whether documents support the correct answer, so we used it as a strong baseline with Flan-UL2 (Tay et al., 2023), as detailed in Section B.2. Interestingly, our evidentiality evaluator approximates (-0.08p%) the performance of Flan-UL2 (20B), our reader LLM, in terms of F1-score, despite its significantly smaller parameter size (770M).

	NQ		TQA	
	EM	**R20**	**EM**	**R20**
(A) ECoRAG (ours)	**44.38**	**75.18**	**66.45**	80.38
Compressor				
(B) w/o answerability	36.51	49.53	64.42	70.84
(C) w/o evidentiality	42.94	74.93	65.59	**80.59**
Adaptive Compression				
(D) w/o evaluator	39.61	-	62.78	-

Table 3: Ablation study of ECoRAG, showing the impact of compressor and adaptive compression methods.

Methods	**Compression Time**	**Inference Time**	**Total Time**	**Throughput (example/sec)**
No Documents	-	3.79h	3.79h	0.26
Raw Documents	-	12.28h	12.28h	0.08
RECOMP	0.27h	4.08h	4.35h	0.23
CompAct	10.10h	4.83h	14.94h	0.07
ECoRAG (ours)	0.73h	4.23h	4.96h	0.20

Table 4: Inference time and compression time for NQ test.

5.3 RQ3: Ablation Study

In Table 3, we report the results of the ablation study to assess the impact of each component in our framework by comparing EM across different settings. We also report R20, which measures whether the gold answer words exist in the top-20 sentences.

For ***Compressor***, we compare (A) ECoRAG with two inferior compressors (B) and (C). In (B), the compressor utilizes a pretrained Contriver checkpoint without additional training. In (C), the compressor is trained with answerability labels. As shown, our compressor trained with evidentiality labels performs better than any other setting. Comparing (A) and (C) shows that evidentiality labels increase EM (+1.44%p) while maintaining R20 at a comparable level. Since R20 measures lexical overlap, (C) which is trained with answerability performs similarity to or better than ours (A). The results demonstrate the superiority of our evidentiality labels over answerability labels, as they prioritize contextually rich information.

For ***Evaluator***, a no-evaluator setting (D) is considered, where the initial compression from the compressor is always used without evaluating its evidentiality. The EM gap between (A) and (D) (+4.77%p) highlights the impact of the evidentiality evaluator. Through comparison, the results highlight the importance of adaptively adjusting the optimal amount of evidence by using the evidentiality evaluator.

5.4 RQ4: Total Latency

ECoRAG is cost-efficient not only because it reduces the number of tokens but also because it decreases total latency in the RAG process. In RAG without compression, computational costs increase as more documents are retrieved. By applying compression and retaining only the necessary information, ECoRAG reduces total processing time.

In Table 4, we measured the total latency, which includes both compression and inference time, to demonstrate the time-efficiency of our approach. For long context, the LLM-based abstractive compressor CompAct took longer than the 'Raw Document' setting, whereas the extractive compressors RECOMP and ECoRAG were faster. ECoRAG uses the lightweight evaluator that only generates a single token per iteration, and reflection process stops once the compressed document is evidential or the token limit is reached, thus preventing excessive compression time. Although ECoRAG had similar speed to RECOMP, it achieved better performance by retaining only the information necessary to generate the correct answer, as described in Table 1. Thus, ECoRAG is effective in long context, both in terms of performance and efficiency.

ECoRAG is a two-step design that achieves both speed and performance. Single-step aggregation with LLMs, as demonstrated by the CompAct in Table 1, struggles with length dependency for list-wise evaluation due to the "lost-in-the-middle" issue (Liu et al., 2024). In contrast, ECoRAG separates the process by first assessing sentences individually with an extractive compressor and evaluating them collectively. This separation overcomes challenges in handling long contexts and enhances compression effectiveness. Our lightweight components maintain efficiency while achieving effective compression.

6 Conclusion

ECoRAG is a framework designed to compress long context by focusing on evidentiality in LLMs, defined as whether information supports generating the correct answer. Evidentiality-guided compression effectively filters out irrelevant content and retains necessary evidence. Through adaptive compression, ECoRAG determines the optimal compression length for each question, ensuring efficient use of context. As a result, ECoRAG demonstrates both superior performance and efficiency in handling long context, outperforming other compression methods.

7 Limitation

Evidentiality provides an effective indicator for determining whether information is necessary for an LLM to generate the correct answer. However, mining evidentiality labels is computationally expensive, leading to increased costs. Since multiple inferences are required for each question, it results in significant time consumption. Nevertheless, as more time is spent, more evidentiality labels can be obtained, which can contribute to the training of the compressor. Evidentiality labels can also be reused to train the evidentiality evaluator, optimizing resource usage. Furthermore, once the compressor is fully trained and applied, the LLM inference process becomes faster.

References

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2022. Evidentiality-guided generation for knowledge-intensive nlp tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2226–2243.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1658–1677, Bangkok, Thailand. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Improving open-domain question answering of LLMs via summarized

retrieval. In *The Twelfth International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Tiziano Labruna, Jon Ander Campos, and Gorka Azkune. 2024. When to retrieve: Teaching llms to utilize information retrieval effectively. *Preprint*, arXiv:2404.19705.

Kyungjae Lee, Seung-won Hwang, Sang-eun Han, and Dohyeon Lee. 2021. Robustifying multi-hop qa through pseudo-evidentiality training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6110–6119.

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.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353, Singapore. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding. *Preprint*, arXiv:2305.04757.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Jesse Mu, Xiang Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Advances in Neural Information Processing Systems*, 36.

Yilin Niu, Fangkai Jiao, Mantong Zhou, Ting Yao, Jingfang Xu, and Minlie Huang. 2020. A self-training method for machine reading comprehension with soft evidence extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3916–3927, Online. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Ruhle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 963–981, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Cheng Qian, Xinran Zhao, and Tongshuang Wu. 2024. "merge conflicts!"' exploring the impacts of external knowledge distractors to parametric knowledge graphs. In *First Conference on Language Modeling*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Yongho Song, Dahyun Lee, Myungha Jang, Seung-won Hwang, Kyungjae Lee, Dongha Lee, and Jinyoung Yeo. 2024. Evidentiality-aware retrieval for overcoming abstractiveness in open-domain question answering. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1930–1943.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *arXiv preprint arXiv:2010.11967*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Kevin Wu, Eric Wu, and James Zou. 2024. How faithful are rag models? quantifying the tug-of-war between rag and llms' internal prior. *arXiv preprint arXiv:2404.10198*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

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

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.

Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. 2024. Compact: Compressing retrieved documents actively for question answering. *arXiv preprint arXiv:2407.09014*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*.

Appendices

A Further Analysis

A.1 Comparative Analysis of Compression Methods

In this section, we will provide a more detailed comparison of our approach with other baselines based on Table 1. Table 5 provides an overview of how each method differs. Based on this comparison, we discuss how large-scale documents can be compressed efficiently and effectively.

In ODQA, since the model must provide an answer to a given question, the compression process needs to consider the question. LLMLingua (Jiang et al., 2023) and LLMLingua-2 (Pan et al., 2024), which do not consider the question during compression, often include irrelevant information, leading to suboptimal performance. On the other hand, the methods other than LLMLingua and LLMLingua-2 are question-aware, allowing them to more effectively capture the necessary content, resulting in higher performance compared to question-agnostic methods.

The amount of evidence needed varies for each question, and one solution to address this is adaptive compression, where the compression rate is adjusted for each question. By applying this method, only the necessary tokens are used, leading to high performance with fewer tokens. As seen in Table 1, both CompAct (Yoon et al., 2024) and ECoRAG achieve high performance with a reduced number of tokens.

However, there are two main challenges when dealing with long context. First, while using numerous retrieval results increases the amount of necessary information available, it also includes documents with lower relevance scores, resulting in considerable noise. Second, the overall length of the documents is too long, which makes the compression process time-consuming.

To address the first challenge mentioned above, the concept of evidentiality is necessary. As discussed in Section 3.1.1, by prioritizing strong evidence for correct answer generation and penalizing distractors, we have been able to create a compressor that is robust against noise. Consequently, this approach allows ECoRAG to demonstrate the highest performance in large-scale document settings.

To address the second challenge, the compressor must be an extractive compressor that evaluates each content pointwise and extracts only the necessary information. Language model-based abstractive compressor is hindered by limited context length, which leads to truncation and fails to handle entire large-scale documents. Moreover, LLM-based abstractive compressor often requires substantial time for inference and may suffer from positional biases (Liu et al., 2024), which can lead to inaccurate assessments of evidentiality. However, extractive compressors such as ECoRAG and RECOMP (extractive) (Xu et al., 2024) are lightweight models that can quickly calculate scores, as seen in Table 4, and process each document in parallel for each document, thus avoiding positional biases.

Based on these observations, we conclude that ECoRAG, which combines all the characteristics from Table 5, is appropriate for compressing large-scale documents effectively.

A.2 Evaluator Performance on NQ

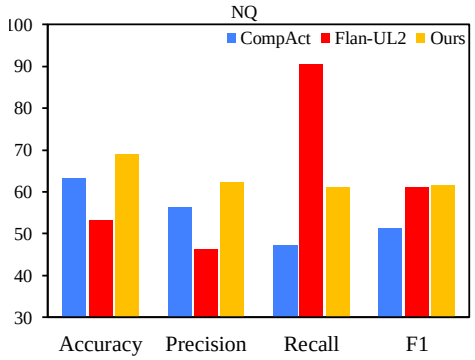


Figure 5: Evidentiality evaluation metrics using different evaluator, including ours, measured on the NQ.

We conducted same experiments on NQ (Kwiatkowski et al., 2019), as described in Section 5.2, observed similar trends to those in TQA (Joshi et al., 2017). As shown in Figure 5, our evidentiality evaluator consistently outperforms CompAct and demonstrates comparable results to Flan-UL2, further validating its effectiveness across different datasets.

A.3 Compression Effectiveness with More Long Context

To explore performance of ECoRAG with more documents, we conducted additional experiments using 1000 retrieved documents in Table 6. Previous compression work, such as CompAct, focused on up to 30 documents, while our experiments used 100 documents, a common setting in RAG models like FiD (Izacard and Grave, 2020). To verify

Methods	Question -aware	Adaptive Compression	Evidentiality -guided	Extractive Compression
LLMLingua, LLMLingua-2	✗	✗	✗	✓
LongLLMLingua	✓	✗	✗	✓
RECOMP (extractive)	✓	✗	✗	✓
RECOMP (abstractive)	✓	✗	✗	✗
CompAct	✓	✓	✗	✗
ECoRAG (ours)	✓	✓	✓	✓

Table 5: The table compares different methods based on their key characteristics. Our approach, ECoRAG, integrates all these features for fast and effective large-scale document compression.

Methods	#tokens ↓	EM	F1
RAG without compression			
closed book	0	21.33	28.71
1000 documents	127,880	0.44	0.63
RAG with 1000 documents compressed			
RECOMP (extractive)	661	31.39	42.29
ECoRAG (ours)	659	**35.51**	**48.63**

Table 6: Experimental results on the NQ test dataset using GPT-4o-mini, comparing performance with and without compression for 1000 documents.

whether our method consistently improves performance even with more documents, we tested with 1000 documents. Due to limited budget, we used documents already retrieved by a DPR setting that was searched, differing from our top-100 DPR setting. Additionally, we tested only RECOMP (extractive), as it is similar to our approach.

With 1000 documents, the context length became too long for GPT-4o-mini to effectively utilize the information (Hsieh et al., 2024). However, our compression effectively reduced the length, maintaining high performance. These results highlight the necessity of compression in managing extended contexts effectively.

A.4 A Comparative Study with Reranker

ECoRAG fundamentally differs from reranking methods like BGE-M3 (Chen et al., 2024) and RECOMP by adaptively determining the rank and compression ratio needed for each query. While reranking models focus on relevance, they lack our ability to iteratively refine compression based on evidentiality. To ensure a fair comparison with our approach in terms of token usage, we conducted additional experiments with BGE-M3 by using its reranked top-10 and top-20 sentences. As shown in Table 7, ECoRAG achieves better performance, demonstrating the importance of selecting the appropriate context over simply increasing or reducing the amount of information.

Unlike other sentence reranking methods, ECoRAG evaluates the initial compression and adaptively adjusts the compression ratio through a reflection process to determine how much information is required. This capability moves ECoRAG closer to true compression rather than simple reranking. Furthermore, our research extends beyond proposing a compressor—it introduces a complete framework. While we used Contriever to ensure fair comparisons with RECOMP, our framework is flexible and capable of training models like BGE-M3 to learn LLM-based evidentiality, further enhancing performance.

A.5 Adaptive Compression Ratio Analysis

To validate the claim of our adaptive compression capabilities, we analyzed the distribution of compression ratios across datasets. The compression ratio is defined as the number of compressed tokens divided by the number of original tokens. Table 8 summarizes the minimum, maximum, mean, median, and standard deviation of compression ratios for the NQ and TQA datasets.

The results highlight differences between datasets, with higher mean and median compression ratios observed for NQ. This reflects complexity of dataset, requiring the extraction of answers from lengthy Wikipedia documents through reasoning and comprehensive understanding. In contrast, TQA involves documents with explicitly positioned answers, making the task primarily about filtering irrelevant information. Consequently, ECoRAG retrieves more evidence for NQ to address its higher information needs, demonstrating its ability to adjust compression ratios adaptively based on dataset complexity and information requirements.

Methods	NQ			TQA			WQ		
	#tokens	EM	F1	#tokens	EM	F1	#tokens	EM	F1
BGE-M3 (top 10)	330	33.02	45.47	370	64.12	74.34	322	20.77	38.27
BGE-M3 (top 20)	670	33.99	46.82	746	65.15	75.14	645	20.77	38.00
ECoRAG (ours)	632	**36.48**	**49.81**	441	**65.34**	**75.37**	560	**30.17**	**46.13**

Table 7: Performance comparison on NQ, TQA, and WQ using GPT-4o-mini, between BGE-M3 (Chen et al., 2024) and ECoRAG.

Dataset	Min Compression Ratio	Max Compression Ratio	Mean Compression Ratio	Median Compression Ratio	Standard Deviation
NQ	0.0036	1	0.0401	0.0446	0.0247
TQA	0.0034	1	0.0267	0.0161	0.0221

Table 8: Compression ratio statistics for NQ and TQA datasets.

A.6 Case study of evidentiality-guided compression

Table 9 illustrates an example of evidentiality-guided compression. For the given question, *who dies at the end of Den of Thieves?* with the correct answer *Merrimen*, the initial document set before compression includes the correct answer. But it also contains irrelevant information, which misleads the LLM into generating the wrong answer, *Donnie.* After compression, irrelevant content containing the word *Donnie* is effectively suppressed, leaving only the evidential (highlighted) sentences.

A.7 Generalizability across Readers

To evaluate the generalizability of our compression framework, we conducted experiments using Flan-UL2 (Tay et al., 2023) (20B), Llama3 (Dubey et al., 2024) (8B), and Gemma2 (Team et al., 2024) (9B) as the reader LLMs. These models were chosen to investigate how our method performs across diverse architectures and parameter sizes.

Flan-UL2 was selected because RECOMP also utilizes it, as we intend to directly compare with it. Furthermore, additional experiments were conducted with Llama3 and Gemma2 to extend the evaluation. Since Llama3 has large context length, it can conduct 'naive prepend' experiment, unlike Flan-UL2 and Gemma2.

Results show that our evidentiality-guided compression method consistently outperforms other compression baselines on all three models. Specifically, with Flan-UL2 in Table 10, which was used to define evidentiality during training, the model demonstrated a clear improvement across all metrics. Similarly, as shown in Table 11. Gemma2, despite being trained without its own evidentiality mining, also showed improved performance with our compression method, further validating its effectiveness.

In the case of Llama3, as presented in Table 12, our compression approach outperformed other baselines, including naive prepend. However, in certain instances, it was outperformed by the 'closed book' approach. This suggests that parametric knowledge embedded within the reader LLM can occasionally align well with specific datasets, leading to variations in performance across models.

Nonetheless, our framework ECoRAG is model-agnostic, as we have excluded the influence of the parametric knowledge of the reader LLM in mining

Question *who dies at the end of den of thieves*		**Gold answers** **Merrimen**
Type	**In-context documents**	**Prediction**
None		**Donnie**
retrieved documents	Den of Thieves (film) Nick, forcing Nick to shoot him. As **Merrimen** lies on the ground dying, Nick kneels and consoles him. When Nick inspects **Merrimen**'s SUV, he only finds bags with shredded paper; he also finds that **Donnie** has escaped custody. Nick later goes to **Donnie**'s bar and sees pictures of him with some of the crew members from the heist. It is revealed **Donnie** masterminded the heist to keep all of the stolen cash for himself in a second garbage truck. After the passage of some time, **Donnie** is working in a London bar, planning a new heist. The film was in Den of Thieves (film) is currently in development. In Los Angeles, a team of robbers led by Ray **Merrimen** make a violent armed attack and hijack an armored truck. Police officers arrive on the scene and engage in a shootout with the robbers. Eventually, **Merrimen** and his crew escape with the empty armored truck. In the morning, Detective Nick O'Brien investigates the crime scene, having been monitoring **Merrimen** and his crew for a while. Suspecting a local bartender named **Donnie** for involvement, Nick finds him at the bar and kidnaps him for interrogation. **Donnie** reveals **Merrimen** is planning to rob the Federal Reserve on Den of Thieves (film) garbage truck that removes shredded bills. Nick's team catches up to **Donnie** and seizes him, beating him until he tells them where **Merrimen** is going. **Merrimen**, Bosco, and Levi try to make their escape with the money bags from the waste truck but hit a traffic jam and are blocked. Nick's team spots them and attempt to shoot them as the robbers try to escape. A shootout occurs initiated by **Merrimen**, killing one of Nick's men. Levi and Bosco are eventually shot dead, but **Merrimen** gets away. Nick chases and shoots **Merrimen**, wounding him. **Merrimen** raises an empty gun to Den of Thieves (film) is currently in development. In Los Angeles, a team of robbers led by Ray **Merrimen** make a violent armed attack and hijack an armored truck. Police officers arrive on the scene and engage in a shootout with the robbers. Eventually, **Merrimen** and his crew escape with the empty armored truck. In the morning, Detective Nick O'Brien investigates the crime scene, having been monitoring **Merrimen** and his crew for a while. Suspecting a local bartender named **Donnie** for involvement, Nick finds him at the bar and kidnaps him for interrogation. **Donnie** reveals **Merrimen** is planning to rob the Federal Reserve on Den of Thieves (film) Friday of that week by covertly removing about $30 million in old bills which are scheduled to be shredded after their serial numbers are deleted from computer records. At their hideout, **Merrimen** has one of his crew, Levi, roughly interrogate **Donnie** to ensure he didn't disclose anything about the plan. Meanwhile, Nick goes to a strip club and finds **Merrimen**'s stripper girlfriend, hiring her for the night to find out where the heist is going to happen. The next morning, Nick makes an effort to see his daughter at her school. As the day of the heist comes, **Merrimen** and	**Donnie**
Compression	Den of Thieves (film) As **Merrimen** lies on the ground dying, Nick kneels and consoles him. Den of Thieves (film) Eventually, **Merrimen** and his crew escape with the empty armored truck. Den of Thieves (film) **Merrimen**, Bosco, and Levi try to make their escape with the money bags from the waste truck but hit a traffic jam and are blocked. Den of Thieves (film) In the morning, Detective Nick O'Brien investigates the crime scene, having been monitoring **Merrimen** and his crew for a while. Den of Thieves (film) Meanwhile, Nick goes to a strip club and finds **Merrimen**'s stripper girlfriend, hiring her for the night to find out where the heist is going to happen.	**Merrimen**

Table 9: Case study of how the compression of the retrieved documents helps the model to identify the correct answer from NQ test set. The highlighted part is the evidential sentence that directly gives useful information for generating the correct answer **Merrimen**, rather than the incorrect answer **Donnie**.

evidentiality labels. These results emphasize that our compression method consistently outperforms other compression approaches, further validating its effectiveness across diverse models and configurations.

Methods	NQ			TQA			WQ		
	#tokens ↓	EM	F1	#tokens ↓	EM	F1	#tokens ↓	EM	F1
RAG without compression									
closed book	0	21.33	28.71	0	46.48	52.47	0	32.97	42.33
100 documents (Karpukhin et al., 2020)	15456	-	-	15943	-	-	15135	-	-
RAG with 100 documents compressed									
LLMLingua	725	19.17	25.48	726	42.97	48.93	868	31.10	40.87
LLMLingua-2	1475	24.63	32.19	1518	53.07	59.42	1580	30.61	41.76
LongLLMLingua	1516	38.03	46.94	1570	65.79	73.88	1629	32.78	45.27
RECOMP (extractive)	727	38.06	46.18	750	62.49	69.68	857	31.25	43.18
RECOMP (abstractive)	**16**	22.22	29.56	**30**	43.50	49.88	**157**	38.15	38.56
CompAct	252	42.16	51.05	253	64.37	72.25	218	33.07	44.45
ECoRAG (ours)	693	**44.38**	**53.56**	501	**66.45**	**74.02**	671	**33.71**	**46.08**

Table 10: Comparison of compression methods on NQ, TQA, and WQ using Flan-UL2 (Tay et al., 2023) with 100 retrieved documents.

Methods	NQ			TQA			WQ		
	#tokens ↓	EM	F1	#tokens ↓	EM	F1	#tokens ↓	EM	F1
RAG without compression									
closed book	0	27.84	38.35	0	57.11	66.39	0	26.77	43.24
100 documents (Karpukhin et al., 2020)	14260	-	-	-	-	-	14075	-	-
RAG with 100 documents compressed									
LLMLingua	643	26.90	37.90	638	60.71	68.09	649	25.04	42.08
LLMLingua-2	1403	28.56	38.95	1393	59.95	67.84	1401	24.36	40.52
LongLLMLingua	1411	37.67	49.40	1436	63.17	70.28	1399	27.02	44.23
RECOMP (extractive)	165	37.65	48.24	687	63.19	70.38	680	26.03	42.22
RECOMP (abstractive)	**17**	27.98	38.00	**28**	58.78	65.74	**21**	25.20	41.60
CompAct	111	38.67	49.87	100	65.88	73.29	78	26.67	43.04
ECoRAG (ours)	684	**39.20**	**50.24**	448	**66.32**	**74.25**	504	**27.41**	**44.00**

Table 11: Comparison of compression methods on NQ, TQA, and WQ using Gemma2 (Team et al., 2024) with 100 retrieved documents.

Methods	NQ			TQA			WQ		
	#tokens ↓	EM	F1	#tokens ↓	EM	F1	#tokens ↓	EM	F1
RAG without compression									
closed book	0	22.16	32.36	0	**60.89**	67.80	0	**21.79**	**35.81**
100 documents (Karpukhin et al., 2020)	14263	0.27	0.97	14574	0.24	2.70	14147	0.25	4.48
RAG with 100 documents compressed									
LLMLingua	641	15.20	22.31	636	52.11	59.23	646	17.62	30.92
LLMLingua-2	1346	3.91	7.19	1366	48.08	55.91	1337	4.28	11.44
LongLLMLingua	1388	20.30	28.85	1423	58.34	68.49	1372	18.70	32.12
RECOMP (extractive)	160	22.33	31.12	683	36.69	44.08	667	16.19	27.80
RECOMP (abstractive)	**16**	18.75	27.85	**27**	42.73	50.94	**21**	18.80	33.25
CompAct	107	28.01	38.52	99	56.01	64.69	76	21.41	35.21
ECoRAG (ours)	519	**30.22**	**42.55**	445	59.25	**69.32**	588	21.60	35.43

Table 12: Comparison of compression methods on NQ, TQA, and WQ using Llama3 (Dubey et al., 2024) with 100 retrieved documents.

B Experimental Details

B.1 Implementation Details

We used 8 Nvidia RTX3090 GPUs to train all models. For mining evidentiality labels for all sentences in retrieved documents, we used the NLTK library [4] to split DPR (Karpukhin et al., 2020) retrieved top-100 documents into sentences. Our evidentiality compressor was trained from a pretrained Contriever (Izacard et al., 2022) checkpoint using AdamW optimizer, with a batch size of 64 for 4 epochs on NQ (Kwiatkowski et al., 2019) and WQ (Berant et al., 2013), and 2 epochs on TQA (Joshi et al., 2017). When calculating the $\mathcal{L}_{se}$ loss, we used negative set with weak evidence to distractor ratio of 0.15:0.75, treating weak evidence as hard negative. We set the temperature τ for the contrastive loss as 1.0.

Our evidentiality evaluator was trained from a pretrained Flan-T5-large checkpoint[5] using the AdamW optimizer, with a batch size of 40 for 4 epochs with all datasets. We included '<NOT>' sentences with high compressor scores in the training stage to make the evidentiality evaluator distinguish only the genuinely strong evidence '<EVI>' from the seemingly plausible ones. We constructed the training data for the evaluator with a ratio of 1:3 between '<EVI>' and '<NOT>' sentences. For adaptive compression, a limit on the number of evidence pieces was necessary to avoid infinite loops, which we set at 20. Additionally, to prevent high latency due to overly frequent evaluations, we incrementally added 4 evidence pieces at a time. For evidential mining, we used Flan-UL2[6] with the 8-bit quantization setting, and for experiments on the test set, we used GPT-4o-mini[7], Flan-UL2, Gemma2[8], and Llama3[9].

B.2 Input Prompts for LLM

We report two examples of input prompts for reader LLMs. In Table 13, we report the input prompt used for evidentiality mining and test set experiments to answer a given question when provided with the question and the compressed documents. This prompt was also utilized during the evidentiality mining process, as described in Section 3.1.1. Table 14 presents the input prompt for mining the ground truth label of compressed documents using Flan-UL2 as the evidentiality evaluator in the experiments detailed in Section 5.2.

C Usage of AI Assistants

We utilized ChatGPT to improve the clarity and grammatical accuracy of my writing. It provided suggestions for rephrasing sentences and correcting grammatical errors to make the text flow more naturally.

[4] https://www.nltk.org/
[5] https://huggingface.co/google/flan-t5-large
[6] https://huggingface.co/google/flan-ul2
[7] gpt-4o-mini-2024-07-18
[8] https://huggingface.co/google/gemma-2-9b-it
[9] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

who won a million on deal or no deal Answer: Tomorrow Rodriguez
who is the woman washing the car in cool hand luke Answer: Joy Harmon
who is the actor that plays ragnar on vikings Answer: Travis Fimmel
who said it's better to have loved and lost Answer: Alfred , Lord Tennyson
name the first indian woman to be crowned as miss world Answer: Reita Faria
`Documents`
`Question`
Answer:

Table 13: Input prompt for to LLM for question answering including few shot examples, input documents, and question.

You are an expert at determining whether a document provides evidential support for a given question. You will receive a question and a document, and your task is to evaluate whether the document is evidential, partially evidential, or non-evidential in relation to the question.
Assess the support provided by the document using the following scale:
- [Evidential] - The document fully supports the question, providing clear and direct evidence that answers or addresses the query completely.
- [Non-Evidential] - The document does not provide relevant information or evidence related to the question, making it unrelated or insufficient to support the query.
Please provide your assessment and briefly justify your reasoning based on the content of the document in relation to the question.

Question: what is the temperature of dry ice in kelvin?
Evidence: At atmospheric pressure, sublimation/deposition occurs at or 194.65 K. The density of dry ice varies, but usually ranges between about.
Score: [Evidential]

Question: when did north vietnam unify with the south?
Evidence: The distinctive synthesizer theme was performed by the then-little-known Thomas Dolby, and this song also marked a major departure from their earlier singles because their previous singles were mid to upper tempo rock songs while this song was a softer love song with the energy of a power ballad.
Score: [Non-Evidential]

Question: who played all the carly 's on general hospital?
Evidence: Throughout the 2000s, Carly, then Tamara Braun (2001–05) goes on to become one of the
Score: [Non-Evidential]

Question: who sang the original blinded by the light?
Evidence: Light of Day (song) "Light of Day", sometimes written as "(Just Around the Corner to the) Light of Day", is a song written by Bruce Springsteen and performed initially by Joan Jett and Michael J.
Score: [Non-Evidential]

Question: who was the rfc editor until 1998 just provide the family name?
Evidence: Perhaps his most famous legacy is from RFC 760, which includes a robustness principle often called "Postel's law": "an implementation
Score: [Non-Evidential]

Question: `Question`
Evidence: `Compressed Documents`
Score:

Table 14: Input prompt to LLM for evidentiality evaluation including few shot examples, compressed documents, and question.