EXIT: Context-Aware Extractive Compression for Enhancing Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

We introduce **EXIT**, an extractive context compression framework that enhances both the effectiveness and efficiency of retrieval-augmented generation (RAG) in question answering (QA). Current RAG systems often struggle when retrieval models fail to rank the most relevant documents, leading to the inclusion of more context at the expense of latency and accuracy. While abstractive compression methods can drastically reduce token counts, their token-by-token generation process significantly increases end-to-end latency. Conversely, existing extractive methods reduce the latency but rely on independent, non-adaptive sentence selection, failing to fully utilize contextual information. EXIT addresses these limitations by classifying sentences from retrieved documents—while preserving their contextual dependencies—enabling parallelizable, context-aware extraction that adapts to query complexity and retrieval quality. Our evaluations on both single-hop and multi-hop QA tasks show that EXIT consistently surpasses existing compression methods and even uncompressed baselines in QA accuracy, while also delivering substantial reductions in inference time and token count. By improving both effectiveness and efficiency, EXIT provides a promising direction for developing scalable, high-quality QA solutions in RAG pipelines [1].

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Khandelwal et al., 2020) is the task of enhancing Large Language Models (LLMs) responses with relevant external contexts or documents. By grounding answers in evidence, RAG systems have gained much attention for mitigating hallucination issues (Ram et al., 2023; Li et al., 2023b) and improving factual reliability (Jeong et al., 2024; Xia et al., 2024b).

[1]We will make our code publicly available upon acceptance of this paper.

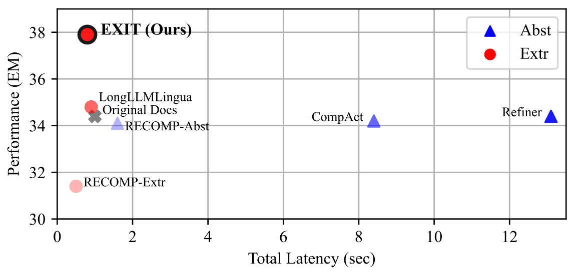

Figure 1: Average QA accuracy (EM) and efficiency (Total Latency) for various compression methods using Contriever-MSMARCO as the retriever and Llama-3.1-8b-Instruct as the reader.

However, they face significant challenges in practical deployment, as retrieval models sometimes fail to rank the most relevant documents at the top (Robertson and Zaragoza, 2009; Izacard et al., 2022). One potential solution is to retrieve a larger set of documents to ensure the coverage of the necessary information, but this approach compromises both effectiveness and efficiency. Specifically, for effectiveness, LLMs often struggle with processing long contexts, overlooking critical information located in the middle of contexts (Liu et al., 2023). Additionally, irrelevant information in retrieved documents can act as distractors, significantly degrading the overall QA performance (Shi et al., 2023a; Li et al., 2023a; Wu et al., 2024). From an efficiency perspective, increasing the context size raises inference latency—due to quadratic complexity in attention computation (Xia et al., 2024a)—and API costs tied to input length[2]. Moreover, the context window limitations inherent in LLM architectures set strict upper bounds on the maximum input size.

To address these challenges, context compression has emerged as a promising solution, condensing essential information from multiple retrieved contexts through either abstractive or extractive approaches. While they can reduce inference time and filter out irrelevant information, both approaches

[2]https://openai.com/api/pricing/

still have notable drawbacks. Specifically, abstractive compression methods—often implemented via autoregressive generation—summarize or rewrite documents into a single condensed passage (Li et al., 2023c; Xu et al., 2024; Yoon et al., 2024; Li et al., 2024; Wang et al., 2023), significantly increasing end-to-end latency due to their token-by-token generation process. For instance, as illustrated in Figure 1, CompAct (Yoon et al., 2024), a representative abstractive approach, takes over 8 seconds to process just five documents for a single query, whereas using the original document without compression takes only 1 second.

On the other hand, extractive compression approaches can offer a more efficient alternative (Xu et al., 2024; Jiang et al., 2023, 2024), by selecting relevant textual segments (e.g., sentences, or even token-level excerpts) directly from retrieved documents. This strategy reduces both compression time and overall latency. However, current extractive methods have yet to reach their full potential in terms of effectiveness (Choi et al., 2021; Pan et al., 2024). They often rely on rigid selection criteria that do not adapt to variations in query complexity or the quality of retrieved documents, and they frequently neglect to fully leverage the broader context when choosing which tokens or sentences to retain. Specifically, as illustrated in Figure 1, while extractive approaches such as RECOMP-Extr (Xu et al., 2024) achieve minimal compression time, their inability to dynamically adjust selection processes results in suboptimal QA performance.

Therefore, in this work, we propose a novel compression framework for RAG, **EX**tract**I**ve Contex**T** compression (EXIT), designed to enhance both effectiveness and efficiency by improving efficiency through an extractive compression strategy and enhancing effectiveness through dynamic, context-aware sentence selection. Specifically, as shown in Figure 2, EXIT operates in three stages: (1) splitting retrieved documents into sentences, (2) performing parallelizable binary classification ("Yes" or "No") on each sentence to assess its relevance while considering its full document context, rather than evaluating them independently, and (3) recombining selected sentences while preserving their original order. Therefore, as shown in Figure 1, EXIT frames context compression as a sentence classification problem, enabling it to outperform both compression methods and the uncompressed baseline in terms of speed. Specifically, it reduces processing time from several seconds to about 1 second. Moreover, by leveraging context-aware and adaptive sentence selection, EXIT also surpasses other extractive methods in accuracy. Also, we note that EXIT operates as a plug-and-play module that can be seamlessly integrated into any existing RAG pipeline without architectural modifications.

We evaluate EXIT on both single-hop QA tasks (NQ (Kwiatkowski et al., 2019), TQA (Joshi et al., 2017)) and multi-hop QA tasks (HQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020)). Experimental results demonstrate that EXIT not only improves effectiveness over both abstractive and extractive compression baselines but also significantly reduces latency compared to abstractive methods and the uncompressed baseline.

Our contributions are as follows:

- We identify and address the key weaknesses of existing context compression methods: abstractive approaches incur prohibitive latency, while traditional extractive methods rely on rigid, non-adaptive content selection.
- We propose **EXIT** (**EX**tract**I**ve Contex**T** Compression), an extractive compression framework that dynamically adjusts to query complexity and retrieval quality.
- We demonstrate, through extensive experiments, that EXIT surpasses previous compression methods and uncompressed retrievals, improving QA performance while significantly reducing both token counts and end-to-end latency.

2 Related Work

Retrieval-Augmented Generation. The RAG pipeline typically follows a naive retrieve-then-generate process, where a single-step retrieval precedes generation (Lewis et al., 2020; Shi et al., 2023b; Ram et al., 2023). However, a simple single-step retrieval often fails to rank relevant documents at the top and struggles to handle multi-hop queries requiring multiple pieces of information. To address these, RAG pipelines have evolved into iterative, recursive, and multi-hop retrieval approaches (Shao et al., 2023; Trivedi et al., 2023; Khattab et al., 2022), which require multiple retrievals for a single query. While these methods improve information coverage, they also increase end-to-end latency from retrieval to generation, reducing the overall efficiency of the pipeline. Moreover,

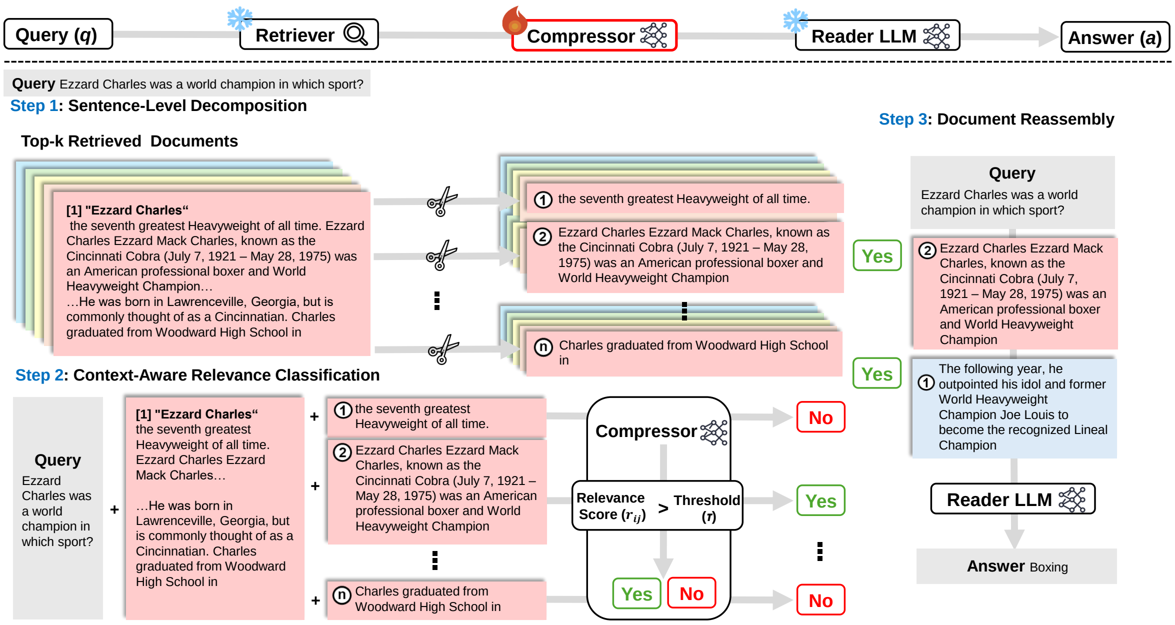

Figure 2: Overview of our framework. First, the retrieved document is split into sentences. Next, each sentence is classified as either "Yes" or "No" using the Compressor. Finally, sentences with scores above the threshold are recombined in their original order to complete the compression.

increasing the document length or the number of retrieved documents as alternative solutions to ensure coverage further exacerbates efficiency issues in these complex retrieval approaches. This not only heightens inference costs (Xia et al., 2024a) but also makes it harder to focus on critical details within the documents (Liu et al., 2023). In response, reducing the tokens in the retrieved documents while keeping key information in them has gotten attention to complement the efficiency issue of the RAG pipeline.

Context Compression. Context compression has emerged as a practical remedy for handling increasingly large prompt lengths in RAG pipelines. Existing approaches commonly fall into soft or hard compression. Soft compression focuses on shortening embedding vectors at the token level (Wingate et al., 2022; Mu et al., 2023; Ge et al., 2024; Chevalier et al., 2023; Cheng et al., 2024), but requires extensive training and architectural changes, making it unsuitable for black-box LLMs.

Hard compression, by contrast, removes non-essential textual content directly (Li et al., 2023d; Jiang et al., 2023), offering a plug-and-play solution compatible even with API-based models such as ChatGPT (OpenAI, 2023). Hard compression techniques are further divided into abstractive and extractive methods. Abstractive methods employ autoregressive models to generate query-focused summaries, thus drastically reducing token counts at the cost of additional latency and potential hallucinations (Zhao et al., 2020). For example, RECOMP-Abst (Xu et al., 2024) uses a T5-based summarizer for token reduction but requires dataset-specific training and slows inference. CompAct (Yoon et al., 2024) and Refiner (Li et al., 2024) take this approach further by leveraging even larger LLMs with 7B parameters, compounding latency issues and increasing resource demands.

Extractive methods select salient segments (e.g., sentences, or tokens) directly from the retrieved documents. This avoids the autoregressive bottleneck and mitigates hallucinations. RECOMP-Extr (Xu et al., 2024) is one such example, but its static and context-agnostic selection of only a few sentences per document limits its performance. Similarly, token-level approaches such as LLMLingua family (Li et al., 2023d; Jiang et al., 2023; Pan et al., 2024; Jiang et al., 2024) can distort semantic coherence by removing key entities or splitting essential facts.

In short, abstractive methods offer strong compression but suffer from latency and potential halluciantions, while extractive methods are often rigid and lack context awareness. Our work addresses these limitations by proposing a parallelizable, context-aware extractive compression framework. It adaptively selects sentences at scale, preserving semantic integrity and efficiently balancing accuracy with speed, even in complex, multi-step retrieval scenarios.

3 Method

In this section, we first present our problem formulation, the RAG pipeline with a compression stage, and our novel compression framework, **EXIT**, which is designed to extract key evidence for answering in a parallel manner.

3.1 Problem Formulation

RAG Pipeline with Compression. Given a query q and a document corpus $\mathcal{C}$, a RAG pipeline first retrieves Top-k relevant document set D:

$$D = \{d_1, \ldots, d_k\} = \texttt{Retriever}(q, \mathcal{C}), \quad (1)$$

The retrieved documents within the document set D are then processed by a compression module that preserves query-relevant information while significantly reducing input length:

$$D' = \texttt{Compressor}(q, D) \text{ s.t. } l(D') \ll l(D), \quad (2)$$

where $l(\cdot)$ represents the function calculating the number of tokens in the document set. After compression, the number of tokens included in D' is substantially decreased compared to D. Finally, an LLM generates the answer a using the compressed set D' and the given query q:

$$a = \texttt{LLM}(q, D'). \quad (3)$$

Objectives of Compression. For effective and efficient compression in the RAG pipeline, three key criteria should be satisfied: (1) D' should contain fewer tokens, as fewer tokens lead to shorter answer generation times (i.e., reading time); (2) D' should retain the essential evidence required to answer the query, ensuring effectiveness; and (3) the compression process should be sufficiently fast enough to avoid significantly increasing the overall end-to-end inference latency.

3.2 Extractive Context Compression (EXIT)

To achieve three objectives of the compression step, EXIT consists of three main components: sentence-level decomposition, context-aware relevance classification, and document reassembly.

Sentence-Level Decomposition. In Step 1 of Figure 2, EXIT divides each retrieved document into individual sentences using a rule-based sentence tokenizer. For each document $d_i \in D$, we produce a sentence set $S_i = \{s_{i1}, s_{i2}, \ldots, s_{in}\}$, where s_{ij} is the j-th sentence in document i. Operating at the sentence level avoids the fragmentation of key phrases and preserves entity relationships that token-level compression techniques (Jiang et al., 2023) often disrupt. As a result, the compressed context preserves both syntactic coherence and semantic integrity, ensuring that key information is effectively retained.

Context-Aware Relevance Classification. To effectively and efficiently filter sentences in D that contain key evidence for answering the question, we design two key components for sentence relevance evaluation: **document consideration** and **single-token prediction**. First, incorporating the entire document d_i is essential, as key evidence may be distributed throughout the context rather than confined to a single sentence, ensuring no critical information is missed and enabling effective context compression. Furthermore, as multiple-token generation proposed in the previous work (Yoon et al., 2024; Li et al., 2024) could compromise the overall efficiency of the RAG pipeline, we design the lightweight relevance calculation with only single-token prediction with “Yes” and “No” from the query q, document d_i, and sentence s_{ij}. In detail, for each candidate sentence s_{ij}, the evaluation model calculates the relevance score r_{ij} of the sentence s_{ij} for a given query q and document d_i as follows:

$$r_{ij} = \frac{P(\text{“Yes”}|q, d_i, s_{ij})}{P(\text{“Yes”}|q, d_i, s_{ij}) + P(\text{“No”}|q, d_i, s_{ij})} \quad (4)$$

where $P(\cdot|\cdot)$ denotes the likelihood of each token from the evaluation model. This relevance calculation is parallelized across multiple sentences, allowing them to be evaluated simultaneously.

Then, among the sentences in D, EXIT selects sentences with a relevance score exceeding a predefined threshold τ, as high relevance scores indicate that a sentence contains critical information. Notably, this selection process results in **an adaptive number of sentences** in the compressed set D', rather than a fixed amount. This adaptive approach aligns with prior work (Jeong et al., 2024), recognizing that the complexity of queries and the amount of key information vary across queries. As a result, our sentence selection strategy enables effective compression while ensuring all key evidence is included in the compressed set.

Document Reassembly. As shown in Step 3 of Figure 2, EXIT reconstructs the compressed document D' by concatenating only the selected sentences in their original order. Following Hwang et al. (2024), preserving the canonical sentence sequence maintains logical flow and contextual coherence. This approach ensures that the resulting compressed document remains understandable and supports accurate downstream reasoning.

3.3 Classifier Model Training

Training Strategy. Our goal is to train a relevance classifier capable of accurately identifying which sentences provide the evidence required to answer a query. To approximate real-world complexity, we utilize a question-answering dataset that requires multi-sentence reasoning and offers explicit sentence-level annotations of essential information. Leveraging these annotations, we model three typical retrieval outcomes: (1) content that directly provides the needed evidence, (2) passages that appear relevant but lack crucial details, and (3) texts that are entirely irrelevant.

Data Sampling. From the annotated dataset, we draw positive examples from sentences explicitly marked as necessary for producing the correct answer. Within the same documents, we select hard-negative examples—sentences that appear related to the query but do not contain the required evidence—thereby simulating plausible but incomplete retrieval scenarios. Additionally, we sample random negatives from unrelated queries, ensuring the classifier learns to dismiss off-topic content. By maintaining a balanced mix of positives, hard negatives, and random negatives, we create a training set that captures a wide spectrum of retrieval conditions.

Training Procedure. Each training instance is represented as (q, s, d, l), where q is the query, s is a candidate sentence, d is the document containing s, and $l \in \{\text{“Yes”}, \text{“No”}\}$ indicates whether s provides the required evidence. We employ a binary cross-entropy loss function to train the classifier:

$$\mathcal{L} = -\mathbb{1}_{l=\text{“Yes”}} \log P(\text{“Yes”}) - (1-\mathbb{1}_{l=\text{“Yes”}}) \log P(\text{“No”}), \tag{5}$$

By exposing the classifier to a balanced and diverse set of retrieval scenarios, we improve its ability to generalize and reliably identify sentences that contain the critical evidence for answering queries.

4 Experiment Setups

We conduct comprehensive experiments to evaluate EXIT's effectiveness and efficiency in context compression for RAG systems. More implementation details of our experiments are in Appendix A.

Datasets. We evaluate on both single-hop and multi-hop question answering datasets: **NaturalQuestions (NQ)** (Kwiatkowski et al., 2019) and **TriviaQA (TQA)** (Joshi et al., 2017) for single-hop QA; **HotpotQA (HQA)** (Yang et al., 2018) and **2WikiMultihopQA (2WIKI)** (Ho et al., 2020) for multi-hop QA. We use the test set for TQA evaluations and development sets for all other datasets. For the train dataset for the classifier, we exploit the train split of HQA, which has relevant annotations to each sentence in the multiple documents for a query.

Model Configuration. Our system consists of three primary components. The retriever employs **Contriever-MSMARCO** (Izacard et al., 2022), a dense retriever fine-tuned on MSMARCO (Nguyen et al., 2016). The EXIT compressor utilizes **Gemma-2B-it** (Mesnard et al., 2024), optimized for efficient parallel processing. For the reader model, we exploit two scales of instruction-tuned models: **Llama3.1-{8, 70}B-Instruct** (Dubey et al., 2024).

Baselines. We compare EXIT against following context compression approaches: **1) Original Documents** serves as uncompressed baseline. For abstractive methods, **2) RECOMP-Abs** (Xu et al., 2024) uses T5-based (775M) summarization tuned for NQ, TQA, HQA (HQA model for 2WIKI), **3) CompAct** (Yoon et al., 2024) implements Mistral-7B-based iterative compression with 5-segment blocks, and **4) Refiner** (Li et al., 2024) uses Llama2-7B-based compression. For extractive methods, **5) RECOMP-Extr** (Xu et al., 2024) employs Contriever-based (110M) sentence-level extraction tuned for NQ, TQA, HQA (HQA model for 2WIKI), and **6) LongLLMLingua** (Jiang et al., 2024) uses Llama2-7B-chat for token-level extraction with 0.4 dynamic compression rate.

Evaluation Metrics. We evaluate our approach using three metrics: **Exact Match (EM)** and **F1** score to measure effectiveness in question answering, and **end-to-end inference latency (Lat.)** in seconds to assess efficiency. Here, end-to-end latency is defined as the total time encompassing both the compression and generation steps, as the retrieval step remains consistent across all methods.

Implementation Details. We conduct retrieval over the December 2018 Wikipedia dump and apply SpaCy for sentence splitting. We employ vLLM v0.5.5 (Kwon et al., 2023) for accelerated inference with the hyperparameter $T = 0.0$ and Top-$P = 1.0$. We empirically set the relevance threshold $\tau = 0.5$. All experiments are conducted on A100-SXM4-80GB GPUs.

Table 1: Performance across models and datasets, measured by EM, F1, and inference latency (Lat.). 8B reader experiments were conducted on a single A100-80GB GPU, while 70B reader experiments utilized 4 A100-80GB GPUs in parallel. Best results for each dataset are highlighted in **bold**, and second best results are underlined. The "Type" column denotes whether a given compressor is abstractive (Abs.) or extractive (Ext.).

Compressor	Type	NQ			TQA			HQA			2WIKI			AVG.		
		EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓
Llama3.1-8B-Instruct																
Original Docs	-	34.6	47.1	1.0	58.8	68.6	0.9	28.1	38.6	1.0	16.1	24.9	1.1	34.4	44.8	1.0
RECOMP-Abst	Abs.	31.3	43.2	1.6	55.9	65.7	1.4	26.5	37.0	2.2	22.7	29.1	2.1	34.1	43.7	1.8
CompAct	Abs.	32.9	44.6	8.5	58.1	67.7	8.8	28.8	39.8	8.3	16.8	26.0	8.1	34.2	44.5	8.4
Refiner	Abs.	32.9	45.0	28.1	59.2	68.9	10.9	28.8	40.0	6.9	16.8	25.4	6.4	34.4	44.8	13.1
RECOMP-Extr	Ext.	34.6	44.6	**0.5**	56.5	65.1	**0.4**	23.4	32.8	**0.4**	11.2	19.6	**0.6**	31.4	40.5	**0.5**
LongLLMLingua	Ext.	30.2	41.5	0.9	59.4	68.0	0.8	28.0	38.0	0.8	21.5	27.4	0.9	34.8	43.7	0.9
EXIT (Ours)	Ext.	**35.9**	**47.8**	0.8	**60.8**	**69.9**	0.7	**30.6**	**41.5**	0.8	**24.2**	**30.8**	0.9	**37.9**	**47.5**	0.8
Llama-3.1-70B-Instruct																
Original Docs	-	35.6	48.0	8.6	65.1	73.9	7.7	33.7	44.5	8.3	20.8	28.3	9.1	38.8	48.7	8.4
RECOMP-Abst	Abs.	34.1	47.0	4.5	61.3	70.6	3.3	30.3	40.8	4.4	24.2	30.3	4.2	37.5	47.2	4.1
CompAct	Abs.	34.1	45.4	11.9	62.6	71.1	11.7	33.8	44.1	11.0	20.5	27.4	11.6	37.8	47.0	11.5
Refiner	Abs.	35.3	47.1	42.5	64.3	73.0	18.3	33.8	44.7	14.6	21.2	28.0	11.2	38.7	48.2	21.6
RECOMP-Extr	Ext.	35.8	45.3	**2.5**	63.5	71.0	**2.2**	27.6	36.7	**2.9**	13.8	19.3	**3.3**	35.2	43.1	**2.7**
LongLLMLingua	Ext.	32.2	44.0	4.4	66.7	75.2	3.9	34.1	45.3	4.0	28.3	**34.8**	4.3	40.3	49.8	4.1
EXIT (Ours)	Ext.	**36.9**	**49.4**	3.9	**67.3**	**75.9**	3.1	**37.0**	**48.3**	3.3	**28.6**	34.5	3.5	**42.5**	**52.0**	3.5

5 Main Results

Table 1 summarizes our evaluation results across multiple datasets and compression strategies. With the 8B reader, EXIT demonstrates strong generalization: although trained solely on HQA, it effectively addresses both single-hop (NQ, TQA) and multi-hop (2WIKI) queries under out-of-domain conditions. Compared to all baseline methods, EXIT consistently improves EM scores—for instance, by 1.3 and 2.0 points on NQ and TQA, and by even larger margins of 2.5 and 8.1 points on HQA and 2WIKI, respectively. Notably, these accuracy gains come with an average latency of just 0.8s, substantially faster than abstractive compression approaches.

The benefits of EXIT become more pronounced at larger scales. Using the 70B reader, EXIT surpasses the accuracy of all competing methods, averaging a 3.7-point improvement in EM and a 3.3-point improvement in F1 over the uncompressed baseline. On HQA, it achieves a 3.3-point EM gain while maintaining an efficient 3.5s latency—faster than using uncompressed documents and still competitive with the previously fastest method, RECOMP-Extr, but with significantly higher accuracy. EXIT's effectiveness and efficiency, especially with larger models, make it a practical solution for large-scale QA applications.

6 Analyses

We conduct a series of analyses examining EXIT's robustness, classification performance, latency factors, and design choices under various configurations. Additional experimental results and analyses are provided in Appendix B.

6.1 Robustness Analysis

To examine EXIT's robustness as the retrieval set size grows, we gradually increased the number of retrieved documents ($k \in \{1, 5, 10, 20, 30\}$) with an 8B reader, as shown in Figure 3. We found that EXIT steadily improves EM scores—from 28.2 points at $k = 1$ to 33.1 points at $k = 30$—while avoiding the performance degradation seen in RECOMP variants and Refiner at high k values. Also, we measured the impact on the efficiency of the RAG pipeline with token counts and end-to-end latency, confirming that EXIT significantly reduces context from 4,497.1 tokens to 594.4 tokens (86.8% fewer) at $k = 30$, even improving the quality. Also, EXIT's latency scales nearly linearly (0.48s to 2.71s) and is much faster than the abstraction methods and the uncompressed baseline. These results demonstrate that EXIT consistently delivers significant accuracy improvements with minimal inference costs, regardless of the number of documents, making it well-suited for tasks involving larger retrieval sets.

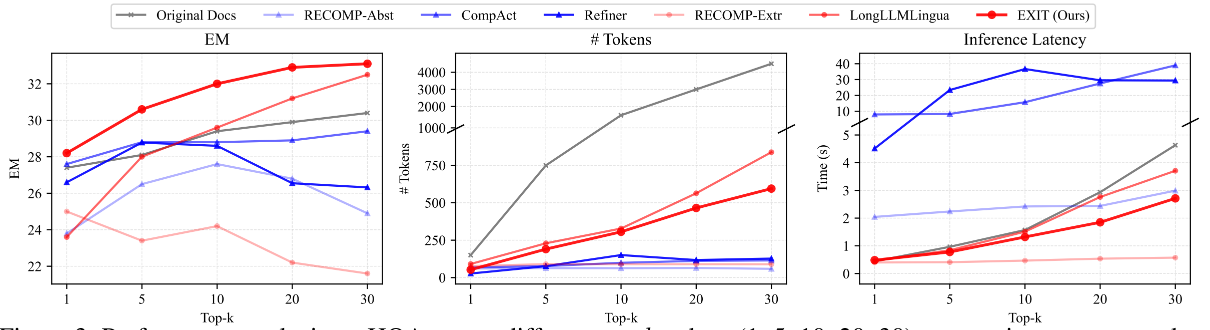

Figure 3: Performance analysis on HQA across different top-k values (1, 5, 10, 20, 30), comparing accuracy, token retention, and inference latency between baselines and our method. All experiments were conducted on a single A100-80GB GPU.

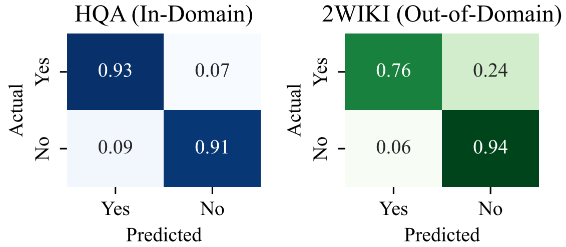

Figure 4: Confusion matrices (row-normalized) for context-aware relevance classification on HQA (in-domain) and 2WIKI (out-of-domain).

6.2 Classification Performance

To better understand the effectiveness of our context-aware relevance classifier, we report row-normalized confusion matrices for both in-domain (HQA) and out-of-domain (2WIKI) datasets, as shown in Figure 4. On HQA, the classifier displays a perfectly balanced ability to recognize both relevant ("Yes") and irrelevant ("No") sentences, achieving over 90% precision and recall in each category. While, on the 2WIKI dataset, the classifier exhibits a slight drop in recall for "Yes" sentences, it still performs strong classification ability with over 70% recall and 90% precision. These results confirm that the classifier performs robustly in its training domain and generalizes reasonably well to unseen queries, yet we leave narrowing this discrepancy as a valuable future research direction.

6.3 Understanding End-to-End Latency Factors

While previous work has primarily focused on minimizing token counts to reduce reading time, we emphasize the importance of considering end-to-end latency, including compression, for building an efficient RAG pipeline. We provide a breakdown of the total end-to-end latency into read time and compression time, along with an analysis of the average number of tokens in compressed documents, as shown in Figure 5. Although some methods

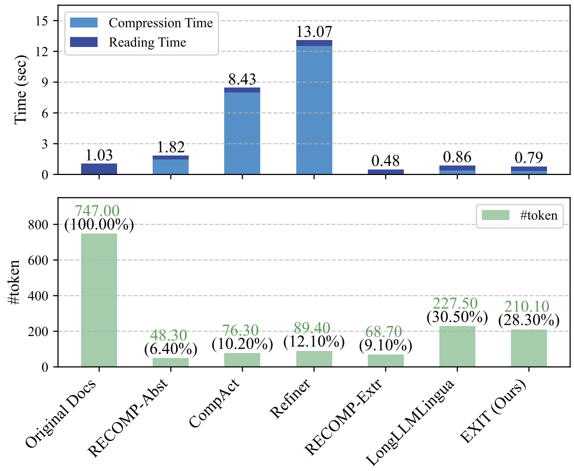

Figure 5: Comparison of compression and reading latency across baselines and our method in QA setting. Experiments were conducted on a single A100 GPU.

achieve extreme token reduction—e.g., RECOMP-Abst (6.4%) and CompAct (10.2%)—their lengthy compression stages (1.46s and 7.99s, respectively) negate these gains, resulting in overall inference times that exceed the uncompressed baseline. By contrast, EXIT retains a moderate token ratio (28.3%) but completes compression rapidly (0.36s), bringing total latency to 0.79s, a 23.3% improvement over the 1.03s needed without compression. This analysis highlights the importance of balancing token reduction and compression inference latency to achieve actual efficiency in the RAG pipeline. Also, our proposed method, EXIT, aligns closely with these objectives, offering a practical approach to context compression for RAG.

6.4 Ablation Studies

To better understand how the design choices in EXIT affect its overall performance and efficiency, we conduct ablation studies focusing on three key components: data sampling strategy, adaptive sentence selection, and context-aware extraction. Table 2 summarizes these results.

Data Sampling Strategy. Our training data com-

Table 2: Ablation studies on HQA examining (1) training data composition (Pos, H-Neg, Neg), (2) adaptive vs. fixed-length sentence selection, and (3) the impact of incorporating passage context during classification.

Configuration	EM ↑	F1 ↑	# token ↓
Ours (Pos + H-Neg + Neg)	**31.6**	**42.6**	195.1
Pos + H-Neg	30.0	41.3	286.8
Pos + Neg	29.8	40.9	404.6
w/o Adaptive Sentence Selection	29.4	40.7	**91.0**
w/o Passage as context	30.4	42.3	157.4

bines positive, hard negative, and random negative samples to mirror the diversity of real-world retrieval scenarios. Compared to using only subsets of these sample types, the comprehensive strategy improves EM by 1.6 points over using only hard negatives and by 1.8 points over only random negatives. Relying solely on positive and random negatives led to excessive token retention, while depending only on hard negatives diminished the model's ability to filter out spurious retrieval noise.

Adaptive Sentence Selection. We compare our adaptive selection mechanism to fixed-length selection. Although fixed-length selection achieves the lowest token count (91.0), it reduces EM by 2.2 points and F1 by 1.9 points. This underscores the importance of adaptively selecting sentences based on the complexity of the query and the retrieved documents, rather than using a static cap.

Context-Aware Extraction. We assess the impact of incorporating full document context when evaluating each sentence's relevance. Removing surrounding context saves 38 tokens but lowers EM by 1.2 points, indicating that broader contextual awareness is crucial for maintaining answer accuracy, even if it slightly increases token count.

These findings confirm that a balanced training data strategy enhances robustness, that adaptive sentence selection ensures efficiency, and that full document consideration preserves accuracy. The classification performances under each ablation setting are reported in Appendix B.5.

6.5 Impact of Compressor Model Size and Compression Strategy

Model Size Considerations. Figure 6 presents an ablation study examining how different base models influence EM scores and total latency. Note that all models trained within EXIT achieve superior accuracy compared to uncompressed baselines and fast compression under 2 seconds. Specifically, Gemma-2B, our base classifier model, achieves a

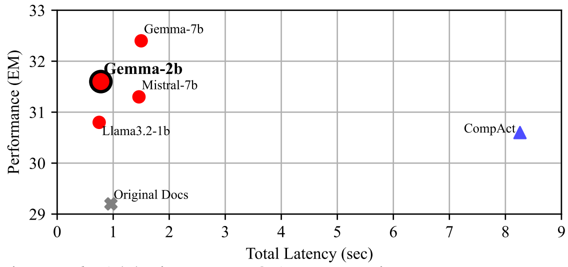

Figure 6: Ablation on HQA comparing EM scores and latency for different model configurations within EXIT (red dot). CompAct and Original Docs are included as indicators. Experiments on a single A100-80GB GPU.

favorable balance between effectiveness and efficiency, delivering 31.6 EM points in 0.78s. When moving to the largest model, Gemma-7B, the highest accuracy (32.4 EM) is achieved, yet it also inflates latency to 1.50s, slightly exceeding the time of uncompressed documents. These results suggest that scaling up model parameters can improve performance but may also compromise latency benefits, emphasizing the flexibility of our proposed framework by selecting an appropriate model depending on the user requirement.

Abstractive vs. Extractive Compression. Figure 6 also compares our extractive approach, EXIT, against CompAct, a 7B-scale abstractive compressor. Using Mistral-7B as the base model for both methods, EXIT (31.3 EM, 1.46s) significantly outperforms CompAct (30.6 EM, 8.26s) in terms of latency and maintains competitive accuracy. This stark difference underscores that the compression strategy, not the just model size, heavily influences efficiency. By relying on extraction rather than iterative summarization, EXIT capitalizes on a large-scale model to preserve high accuracy without incurring prohibitively long inference time.

7 Conclusion

We present EXIT, an efficient context compression framework for RAG systems that leverages parallel processing and context-aware, adaptive sentence selection. Our experiments demonstrate that EXIT achieves superior performance across both single-hop and multi-hop QA tasks while maintaining practical inference speeds. Despite being trained only on HQA, EXIT shows a strong zero-shot generalization ability and proves effective across a wide range of open-source models of varying sizes. These results suggest that efficient parallel extraction with smaller models can outperform larger abstractive approaches, offering a practical solution for real-world RAG applications.

Limitation

Our current approach relies on explicit sentence-level annotations to train the classifier. While these annotations were obtained manually in our experiments, they could potentially be automated through alternative means, such as by GPT-4 supervision or signals derived from the reader itself. We have not yet explored these automated annotation strategies, but doing so remains a promising avenue for future work. Additionally, our study primarily focuses on a general-domain setting, leaving questions about the classifier's performance in specialized domains unanswered. Investigating how well our approach generalizes to domain-specific or highly specialized corpora presents another valuable direction for future research. Lastly, we focus on a single-step RAG pipeline, where retrieval occurs only once, excluding more complex pipelines (Shao et al., 2023; Trivedi et al., 2023; Khattab et al., 2022). However, our proposed framework, EXIT, is orthogonal to these approaches and can be seamlessly integrated by compressing the retrieved documents from each retrieval step.

Ethics Statement

This work enhances RAG-based QA without generating new content beyond what is retrieved. However, biases and inaccuracies in the source documents can still propagate through our compression process. Ensuring the reliability, fairness, and proper curation of underlying corpora is essential for ethical deployment. Future efforts should integrate bias detection, provenance tracking, and user-centric evaluations to promote more transparent and equitable real-world applications.

Acknowledgments

References

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. *CoRR*, abs/2405.13792.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3829–3846. Association for Computational Linguistics.

Eunsol Choi, Jennimaria Palomaki, Matthew Lamm, Tom Kwiatkowski, Dipanjan Das, and Michael Collins. 2021. Decontextualization: Making sentences stand-alone. *Trans. Assoc. Comput. Linguistics*, 9:447–461.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and

Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv.2403.14403*, abs/2403.14403.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13358–13376. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 1658–1677. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix X. Yu, and Sanjiv Kumar. 2023a. Large language models with controllable working memory. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1774–1793. Association for Computational Linguistics.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023b. Halueval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6449–6464. Association for Computational Linguistics.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023c. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6342–6353. Association for Computational Linguistics.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023d. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 6342–6353. Association for Computational Linguistics.

Zhonghao Li, Xuming Hu, Aiwei Liu, Kening Zheng, Sirui Huang, and Hui Xiong. 2024. Refiner: Restructure retrieval content efficiently to advance question-answering capabilities. *CoRR*, abs/2406.11357.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *arXiv:2307.03172*, abs/2307.03172.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth

Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

Jesse Mu, Xiang Li, and Noah D. Goodman. 2023. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

OpenAI. 2023. GPT-4 technical report. *arXiv:2303.08774*, abs/2303.08774.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, 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, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 963–981. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Trans. Assoc. Comput. Linguistics*, 11:1316–1331.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

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, Singapore, December 6-10, 2023*, pages 9248–9274. Association for Computational Linguistics.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023a. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023b. REPLUG: retrieval-augmented black-box language models. *arXiv.2301.12652*, abs/2301.12652.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10014–10037. Association for Computational Linguistics.

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

David Wingate, Mohammad Shoeybi, and Taylor Sorensen. 2022. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5621–5634. Association for Computational Linguistics.

Zhenyu Wu, Chao Shen, and Meng Jiang. 2024. Instructing large language models to identify and ignore irrelevant conditions. *arXiv.2403.12744*, abs/2403.12744.

Heming Xia, Zhe Yang, Qingxiu Dong, Peiyi Wang, Yongqi Li, Tao Ge, Tianyu Liu, Wenjie Li, and Zhifang Sui. 2024a. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 7655–7671. Association for Computational Linguistics.

Peng Xia, Kangyu Zhu, Haoran Li, Hongtu Zhu, Yun Li, Gang Li, Linjun Zhang, and Huaxiu Yao. 2024b. RULE: reliable multimodal RAG for factuality in medical vision language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1081–1093. Association for Computational Linguistics.

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*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.

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

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. *CoRR*, abs/2009.13312.

Appendix

In the Appendix, we provide additional implementation details and present supplementary results and analyses not covered in the main text.

A More Implementation Details

This section describes our training environment, data composition, and prompt templates. All experiments were conducted on an NVIDIA A100-SXM4-80GB GPU cluster. Training was performed on a single GPU with gradient accumulation.

A.1 Training Configuration

We trained the compressor model using the following settings:

- Batch size: 8 per device
- Gradient accumulation steps: 8
- Learning rate: 1e-5
- Weight decay: 0.1
- Warmup ratio: 0.03
- Training epochs: 1

Memory Optimization:

- Optimizer: `paged_adamw_8bit`
- Quantization: 4-bit with float16
- LoRA configuration: Rank=64, Scaling=32, Dropout=0.05

A.2 Model Selection and Training Time

Model selection was based on validation loss. Training required approximately 90 hours on our cluster.

A.3 Data Processing

Table 3: Statistics of the training dataset constructed from HQA. Positive (Pos) sentences are required for the correct answer, Hard-Neg (H-Neg) sentences appear in the same passages but lack crucial evidence, and Neg sentences come from unrelated queries. Counts are in thousands (K).

Split	Pos	H-Neg	Neg	Total
Train	213K	107K	107K	427K
Valid	2.4K	1.2K	1.2K	4.8K

We used SpaCy to segment documents into sentences. Table 3 shows the composition of the training and validation sets derived from HQA. The training set contains 427K sentences, including 213K positive (Pos), 107K hard-negative (H-Neg), and 107K negative (Neg) instances. The validation set includes 4.8K sentences with a similar distribution. This balanced composition ensures the classifier encounters diverse retrieval scenarios during training.

A.4 Inference Settings

For inference, we set:

- Temperature: 0.0
- Top-p: 1.0
- vLLM version: v0.5.5
- Relevance threshold (τ): 0.5

A.5 Prompt Templates

Table 4: A prompt template for document compression.

Compression Prompt Template
Query: {query} Full context: {original passage} Sentence: {sentence} Is this sentence useful in answering the query? Answer only “Yes” or “No”.

Table 5: A prompt template used by the reader model for the QA task.

QA Prompt Template
Context information is below. --- {context} --- Given the context information and not prior knowledge, answer the query. Do not provide any explanation. Query: {query} Answer:

Full prompt templates for compression and QA tasks are provided in Tables 4 and 5, respectively.

A.6 Reproducibility

We will release our codebase, including dataset preprocessing scripts, evaluation protocols, and model checkpoints, upon publication. All random seeds are set to 42 to facilitate reproducibility.

Table 6: Performance comparison between in-domain (HQA) and out-of-domain (2WIKI) datasets using BM25 as the reader model. Best results are highlighted in **bold**, and second best results are underlined.

Compressor	Type	HQA			2WIKI		
		EM ↑	F1 ↑	# Token ↓	EM ↑	F1 ↑	# Token ↓
Top-5 Documents							
Original Docs	-	28.2	39.1	755.8 (100.0)	19.6	25.9	789.7 (100.0)
RECOMP-Abst	Abs.	27.8	39.2	**63.0 (8.3)**	**25.0**	**30.6**	**55.7 (7.1)**
CompAct	Abs.	30.6	41.2	76.9 (10.2)	19.6	29.1	71.0 (9.0)
Refiner	Abs.	30.8	42.3	84.0 (11.1)	19.6	27.8	62.9 (8.0)
RECOMP-Extr	Ext.	28.2	37.5	93.1 (12.3)	11.8	19.1	99.1 (12.5)
LongLLMLingua	Ext.	28.6	39.7	230.3 (30.5)	22.2	27.4	236.6 (30.0)
EXIT (Ours)	Ext.	**33.4**	**44.5**	177.8 (23.5)	24.4	29.1	138.2 (17.5)
Top-20 Documents							
Original Docs	-	31.2	42.5	3009.5 (100.0)	23.0	30.6	3132.5 (100.0)
RECOMP-Abst	Abs.	29.0	39.7	**69.7 (2.3)**	22.8	28.2	**52.3 (1.7)**
CompAct	Abs.	31.6	43.0	109.5 (3.6)	20.4	28.7	113.2 (3.6)
Refiner	Abs.	28.4	38.8	136.1 (4.5)	18.8	26.9	108.4 (3.5)
RECOMP-Extr	Ext.	28.4	37.0	93.5 (3.1)	11.2	19.3	96.7 (3.1)
LongLLMLingua	Ext.	31.0	40.7	558.0 (18.5)	23.6	28.3	593.7 (19.0)
EXIT (Ours)	Ext.	**35.2**	**46.9**	411.3 (13.7)	**27.2**	**32.4**	312.7 (10.0)

Table 7: Performance comparison between in-domain (HQA) and out-of-domain (2WIKI) datasets using GPT-4o as the reader model. Best results are highlighted in **bold**, and second best results are underlined.

Compressor	Type	HQA			2WIKI		
		EM ↑	F1 ↑	#token (%) ↓	EM ↑	F1 ↑	#token (%) ↓
Top-5 Documents							
Original Docs	-	37.2	48.6	735.3 (100.0)	31.2	35.3	764.5 (100.0)
RECOMP-Abst	Abs.	29.4	40.2	**62.8 (8.5)**	23.8	27.8	**53.5 (7.0)**
CompAct	Abs.	37.4	48.0	74.3 (10.1)	30.0	33.6	67.6 (8.8)
Refiner	Abs.	35.8	47.5	71.4 (9.7)	27.8	32.6	54.2 (7.1)
RECOMP-Extr	Ext.	32.4	41.7	87.1 (11.8)	25.6	28.6	93.6 (12.2)
LongLLMLingua	Ext.	34.2	45.2	223.1 (30.3)	30.6	34.5	230.5 (30.2)
EXIT (Ours)	Ext.	**38.2**	**50.4**	191.2 (26.0)	**31.8**	**35.8**	145.6 (19.0)
Top-20 Documents							
Original Docs	-	**39.6**	**51.8**	2940.5 (100.0)	**40.0**	**43.8**	3066.2 (100.0)
RECOMP-Abst	Abs.	33.6	44.2	**62.7 (2.1)**	26.6	32.1	**48.9 (1.6)**
CompAct	Abs.	33.0	43.7	106.0 (3.6)	23.0	27.3	105.1 (3.4)
Refiner	Abs.	31.8	41.5	130.6 (4.4)	31.0	35.5	100.3 (3.3)
RECOMP-Extr	Ext.	31.2	39.6	86.2 (2.9)	23.2	27.2	91.0 (3.0)
LongLLMLingua	Ext.	38.8	49.4	549.5 (18.7)	35.4	39.6	581.5 (19.0)
EXIT (Ours)	Ext.	39.4	50.1	453.6 (15.4)	35.6	40.3	346.7 (11.3)

B Additional Experimental Results and Analyses

Table 13 shows detailed results for each dataset and model configuration under zero-shot QA prompts, using both Top-5 and Top-20 retrieval. For the few-shot QA prompts, Table 14 summarizes the results, where we randomly selected five training examples per dataset as demonstrations. Table 15 provides comprehensive token count statistics, and Table 16 breaks down end-to-end latency.

B.1 Performance with Sparse Retrieval

To assess EXIT's robustness with different retrieval architectures, we evaluate it with BM25, a sparse retrieval method. Table 6 compares EXIT's performance on HQA (in-domain) and 2WIKI (out-of-domain) under Top-5 and Top-20 retrieval settings.

With Top-5 retrieval, EXIT shows notable gains on HQA, improving EM (33.4 vs. 28.2) and F1 (44.5 vs. 39.1) over the uncompressed baseline. Although RECOMP-Abst performs best on 2WIKI (25.0 EM, 30.6 F1), EXIT remains competitive (24.4 EM, 29.1 F1).

EXIT's advantages grow with Top-20 retrieval. On HQA, EXIT outperforms all baselines, improving EM by 4.0 points (35.2 vs. 31.2) and F1 by 4.4 points (46.9 vs. 42.5) compared to using uncompressed documents. On 2WIKI, EXIT achieves the highest scores (27.2 EM, 32.4 F1), confirming its generalizability across domains and retrieval strategies.

B.2 Performance with Proprietary Model

We further examine EXIT's effectiveness using GPT-4o as the reader. Table 7 compares performance on HQA (in-domain) and 2WIKI (out-of-domain), along with compression rates.

For Top-5 retrieval, EXIT attains the best accuracy on HQA (38.2 EM, 50.4 F1) while retaining only 26.0% of tokens. This surpasses the uncompressed baseline (37.2 EM, 48.6 F1) with a 74% token reduction. On 2WIKI, EXIT maintains leading accuracy (31.8 EM, 35.8 F1) while using just 19.0% of the original tokens.

Under Top-20 retrieval, where uncompressed documents benefit from greater coverage, EXIT still achieves competitive accuracy with substantially fewer tokens. On HQA, EXIT closely matches the uncompressed EM score (39.4 vs. 39.6) while using only 15.4% of tokens. Although RECOMP variants compress more aggressively, they suffer marked performance drops. LongLLMLingua performs similarly to EXIT but retains more tokens (18.7% vs. 15.4%).

These findings illustrate EXIT's ability to balance performance and efficiency, making it valuable for API-based proprietary models where token costs and accuracy both matter.

B.3 Impact of Threshold τ

We analyze EXIT's sensitivity to the relevance threshold τ. Figure 7 shows EXIT's performance across various τ values.

EXIT remains stable over a wide threshold range, with strong results between τ=0.3–0.5. At τ=0.3,

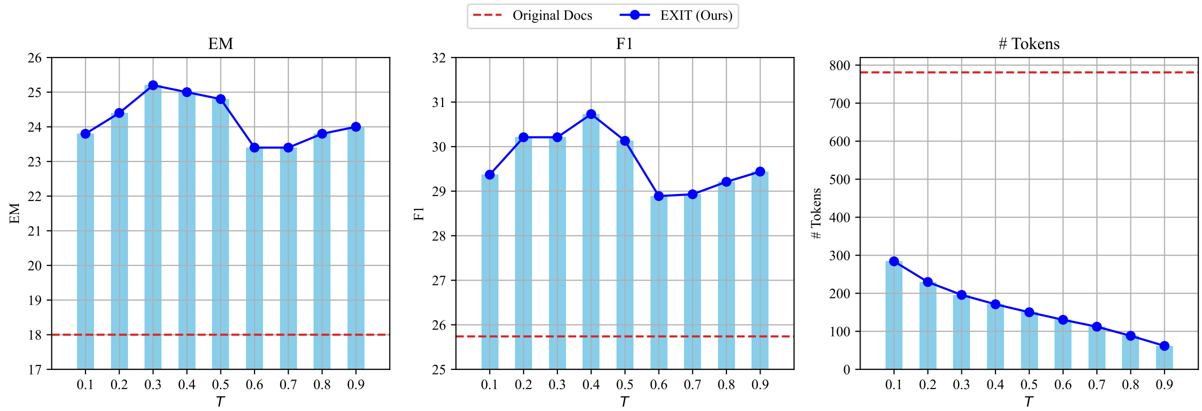

Figure 7: Changes in EM, F1 score, and token count as the threshold T for retaining sentences is adjusted.

Table 8: Classification performance for Yes/No labels, including overall accuracy.

Class	Precision ↑	Recall ↑	F1-Score ↑
Overall			
Yes	0.91	**0.93**	**0.92**
No	**0.93**	0.91	**0.92**
Hard Negative			
Yes	0.86	**0.93**	**0.89**
No	**0.93**	0.84	0.88
Negative			
Yes	**0.96**	0.93	**0.95**
No	0.93	**0.96**	**0.95**

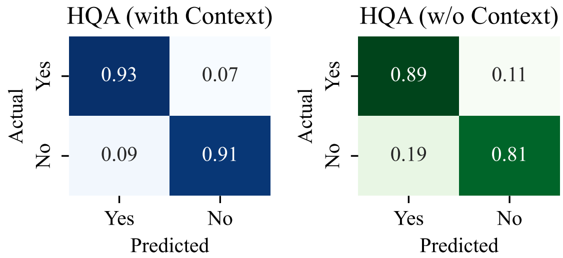

Figure 8: Row-normalized confusion matrices comparing classification performance with (left) and without (right) contextual information on HQA. The availability of context improves the model's ability to accurately distinguish relevant ("Yes") from non-relevant ("No") sentences.

EXIT reaches 25.2 EM using only 25% of the tokens (195.82 vs. 780.95 for the baseline), a substantial improvement over the original documents (18.0 EM). F1 scores also remain consistently higher than the baseline (30.21–30.73 vs. 25.74).

Even under extreme compression (τ=0.9, 7.9% of tokens), EXIT achieves better accuracy (24.0 EM, 29.44 F1) than the uncompressed documents. Conversely, a lenient threshold (τ=0.1) retains more tokens but still provides benefits, demonstrating that EXIT effectively identifies crucial content under varying conditions.

This robustness across thresholds gives practitioners flexibility to adjust the compression-accuracy trade-off without severely impacting performance.

B.4 Analysis of Classification Performance Across Negative Sample Types

Table 8 presents EXIT's sentence-level classification performance, broken down by negative sample type. EXIT achieves 0.92 F1 for both positive ("Yes") and negative ("No") classes overall, indicating a balanced ability to identify essential and non-essential sentences.

The model excels at filtering random negatives (0.95 F1), effectively discarding irrelevant content. With hard negatives (topically related but not essential), EXIT still performs well (0.89 F1 for positive, 0.88 for negative), handling nuanced relevance distinctions.

These results highlight EXIT's adaptability and confirm its suitability for real-world RAG scenarios where both overtly irrelevant and subtly extraneous content must be managed.

B.5 Classification Performance under Ablation Setting.

B.5.1 Analysis of Training Data Composition

Figure 9 presents row-normalized confusion matrices comparing classification performance across three training data configurations: Ours (Pos+H-Neg+Neg), Pos+H-Neg, and Pos+Neg. Under the Ours setup, the classifier displays a balanced ability to identify both "Yes" (relevant) and "No" (irrelevant) sentences, achieving an F1-score of 0.92

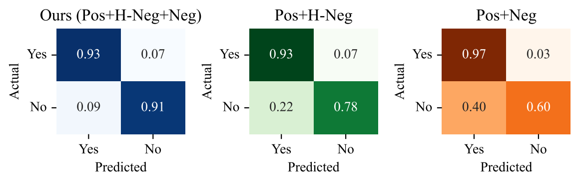

Figure 9: Row-normalized confusion matrices for classification performance under different training data conditions: Ours (Pos+H-Neg+Neg), Pos+H-Neg, and Pos+Neg. Each matrix compares the predicted (“Yes”/“No”) labels against the actual labels.

for both classes. In contrast, excluding one type of negative sample (Pos+H-Neg or Pos+Neg) reduces overall robustness, evidenced by declines in both accuracy and class-wise F1 scores. For instance, the Pos+Neg configuration struggles to maintain balance, accurately identifying “Yes” instances but misclassifying a substantial number of “No” cases. These results confirm that incorporating a comprehensive mix of positive, hard-negative, and random-negative samples leads to more reliable and contextually aware sentence selection, thereby improving the classifier’s performance in practical retrieval-augmented QA scenarios.

B.5.2 Impact of Context on Classification Performance

We evaluate the classifier’s performance with and without broader passage-level context. Figure 8 shows that including context maintains over 90% precision and recall for both “Yes” and “No” classes. Without context, precision and recall decline, weakening the distinction between relevant and irrelevant sentences. This emphasizes the importance of incorporating passage-level context for accurately identifying answer-critical information.

B.6 Training Data Ablation Analysis

Table 9 compares models trained on HQA, 2WIKI, or both. Training solely on HQA yields the highest EM and F1 (31.6 EM, 42.6 F1) with moderate token usage. In contrast, 2WIKI training improves compression but lowers accuracy (29.2 EM, 40.3 F1). Combining datasets does not surpass HQA alone.

This finding suggests that data quality and structure matter more than quantity. HQA’s annotations appear particularly effective for learning robust compression strategies that generalize well, validating our choice to use it as the primary training dataset.

Table 9: Training data ablation comparison. Best results per metric are highlighted in **bold**.

Training Data	EM ↑	F1 ↑	# token ↓
HQA	**31.6**	**42.6**	195.1
2WIKI	29.2	40.3	**135.3**
2WIKI+HQA	30.6	42.0	232.2

B.7 Case Studies

To illustrate how EXIT’s extractive compression strategy improves both accuracy and readability, we present qualitative examples and comparisons with other compression methods.

Original Documents vs. Ours. In Table 10, the original documents contain the correct answer (“Custard”) but also include distracting information (“Eggnog”). Despite having the necessary evidence, the reader fails to produce the correct answer, likely due to this distractor. In contrast, our method (Ours) filters out irrelevant details, drastically reduces input length, and retains only the essential context needed to answer the query accurately. As a result, the reader confidently generates the correct answer (“Custard”).

In a second scenario (Table 11), the original documents retrieve multiple documents related to “Wagner” or “sci-fi” series but fail to provide any content explicitly linking James Belushi to the correct 90s sci-fi series, resulting in an incorrect prediction. Surprisingly, Ours removes all retrieved context entirely, providing the reader with no additional information. Under this no-context condition, the reader relies solely on its internal knowledge and, in this case, correctly identifies “Wild Palms.” While this outcome indicates a form of hallucination or model bias—since the answer emerges without external supporting evidence—it also demonstrates Ours’ capability to avoid misleading context. By eliminating irrelevant or confusing documents, Ours can sometimes allow the model’s internal knowledge to surface, leading to correct answers even in the absence of any retrieved information.

Comparisons with Other Methods. Table 12 compares Ours with several competing compression approaches. CompAct preserves some relevant information but introduces hallucinations, causing the reader to claim that it cannot find the correct answer. Refiner omits the crucial entity required to answer the query, demonstrating how abstractive compressors may inadvertently remove key content.

In contrast, Ours avoids hallucinations and retains the answer's entity in a concise, coherent form.

LongLLMLingua's token-level filtering approach yields unreadable text and removes the essential "Romania" entity, preventing the reader from generating the correct answer. Ours, on the other hand, maintains semantic coherence and includes the correct entity, allowing the reader to produce the correct answer without interference.

These case studies highlight the advantages of Ours: it eliminates distractors, preserves critical entities, and maintains semantic integrity. Consequently, the reader consistently arrives at correct answers with fewer tokens and no hallucinations.

Table 10: Case study comparing compressed contexts and answers between Original Docs and Ours.

	Original Docs	**Ours**
Query	Crème Anglaise is the French version of which English dessert item?	
Context	[1] Creme anglaise Crème anglaise Crème anglaise (French for ""English cream"") is a light pouring Custard used as a dessert cream or sauce. It is a mix of sugar, egg yolks, oil, and hot milk ... [2] Creme anglaise However, the ice cream base is much thicker and has various flavourings. The American South it is known as ""Custard."" It can be served like Eggnog during the Christmas season. Other names include the French terms ""crème à l'anglaise"" (""English-style cream"") and ""crème française"" (""French cream""). Crème anglaise Crème anglaise (French for ""English cream"") is a light pouring Custard used as a dessert cream or sauce. It is a mix of sugar, egg yolks, oil, and hot milk often flavoured with vanilla. Its name may derive from the prevalence of sweet Custards in English desserts. The cream is made by [3] Creme anglaise ... [4] Custard lemon. ""Crème pâtissière"" is a key ingredient in many French desserts including mille-feuille (or Napoleons) and filled tarts. It is also used in Italian pastry and sometimes in Boston cream pie. The thickening of the Custard is caused by the combination of egg and cornstarch. Corn flour or flour thicken at 100 °C and as such many recipes instruct the pastry cream to be boiled. In a traditional Custard such as a ""crème anglaise"", where egg is used alone as a thickener, boiling results in the over cooking and subsequent 'curdling' of the Custard; however, in a pastry cream, starch [5] Cremeschnitte usually pure thick Custard, less commonly combined with meringue (whipped egg whites and sugar) creme. A similar recipe with only meringue filling is called Šampita. In Australia, the dish is more commonly known as a 'vanilla slice'. Cremeschnitte A cremeschnitte („„„„) is a chantilly and Custard cream cake dessert popular in several Central European countries. There are many regional variations, but they all include a puff pastry base and Custard cream. In Slovenia, kremna rezina is commonly associated with the town of Bled, an Alpine tourist destination in northwestern Slovenia. The recipe	[1] Creme anglaise Crème anglaise Crème anglaise (French for ""English cream"") is a light pouring Custard used as a dessert cream or sauce. [2] Creme anglaise Other names include the French terms ""crème à l'anglaise"" (""English-style cream"") and ""crème française"" (""French cream""). Crème anglaise Crème anglaise (French for ""English cream"") is a light pouring Custard used as a dessert cream or sauce. [3] Creme anglaise Alternatively, it can be drunk as a dessert on its own, for example in ""Île flottante"" (""floating island""): the cream is poured into a bowl with a piece of meringue (""blancs en neige"") floated on top along with praline. It can also be used as a base for desserts such as ice cream or crème brûlée. [4] Custard ""Crème pâtissière"" is a key ingredient in many French desserts including mille-feuille (or Napoleons) and filled tarts.
Answer	Custard	
Predict	Eggnog	Custard

Table 11: Case study comparing compressed contexts and answers between Original Docs and Ours. Despite containing no relevant context, Ours method predicts the correct answer, indicating a hallucination scenario.

	Original Docs	**Ours**
Query	Which 90s sci fi series with James Belushi was based on Bruce Wagner's comic strip of the same name?	
Context	[1] Michael I. Wagner patterned his character after Wagner's mannerisms and physical behavior. The series ran on Thursday nights in the Spring of 1988 during the same time slot as NBC's "The Cosby Show", and with that competition could not attract a sufficient audience to get renewed for the following season. ... Wagner helped develop and write the Bochco animated series "Capitol Critters", he also wrote and served as supervising producer [2] Michael I. Wagner Steven Bochco and several of his projects. Wagner was asked by ABC in 1987 to help develop a new science fiction series, " Probe ", a light-hearted series about a scientific crime fighter named Austin James. ... Parker Stevenson, who played the lead character, stated in a later interview that he [3] John Wagner the mid-1990s Wagner worked on a number of licensed properties for Dark Horse Comics in the US, including "Aliens", "Star Wars" – notably solo stories starring Boba Fett and the comics strand of the multimedia project "" – and "". ... It was nominated for the Angoulême International Comics Festival Prize for Scenario in 2006. In 2000 Wagner [4] Martin Wagner (artist) Martin Wagner (artist) Martin Wagner (born April 27, 1966) is an American artist, cartoonist, and filmmaker. ... . His production schedule became increasingly protracted and he ceased publishing the series altogether following issue No. 12 in 1994. In 1996 he made a [5] Wired (film) "L.A. Law", "Murphy Brown", and "Seinfeld"), Chiklis gained fame for portraying the lead roles of Commissioner Tony Scali on the ABC police drama "The Commish" (1991-1996), and LAPD Detective Vic Mackey on the FX police drama "The Shield" (2002-2008) and as well as Marvel superhero Ben "The Thing" Grimm in the films "Fantastic Four" (2005) and "" (2007). ... Wired (film) Wired is a 1989 biographical film of comedian and actor John Belushi, directed by Larry Peerce, and adapted from the 1984 book of the same name by "Washington"	*No context* (completely pruned)
Answer	Wild Palms	
Predict	Probe	Wild Palms

Table 12: Combined case studies comparing compressed contexts and answers across different methods.

Case Study 1	**CompAct**	**Ours**
Query	Where was there a series of explosions and fire on 23 October 1989 (killing 23 employees and injuring 314), an explosion on 23 June 1999 (killing two men and injuring three), and an explosion and fire on 27 March 2000 (causing 1 death and 71 injuries)?	
Context	The Flixborough disaster was an explosion at a chemical plant near Flixborough, England on 1 June 1974. The incident started on October 23, 1989, at 1400 Jefferson Road, Pasadena, Texas .	[1] Phillips disaster of 1989 On October 23, 1989, at 1400 Jefferson Road, Pasadena, Texas , a powerful explosion killed 23 and injured 314. A powerful and devastating explosion and fire ripped through the HCC, killing 23 persons—all working at the facility—and injuring 314 other (185 Phillips 66 employees and 129 contract employees).
Answer	Pasadena, Texas	
Predict	I couldn't find any information about the specific incidents you mentioned. The Flixborough disaster occurred on 1 June 1974.	Pasadena, Texas
Case Study 2	**Refiner**	**Ours**
Query	Who said that there are only three true sports, bullfighting, mountaineering, and motor racing, the rest are merely games?	
Context	1.1. Extreme sport There are only three sports: bullfighting, motor racing, and mountaineering; all the rest are merely games.	[1] Extreme sport usually, but wrongly, attributed to Ernest Hemingway . The phrase is: There are only three sports: bullfighting, motor racing, and mountaineering; all the rest are merely games.
Answer	Ernest Hemingway	
Predict	There is no information provided about who said that.	Ernest Hemingway
Case Study 3	**LongLLMLingua**	**Ours**
Query	Which was the only Eastern Bloc country to participate in the 1984 LA Olympics?	
Context	984 Summer Olympicscottthe majority of Soviet Bloc will not particip Games,ausescu's Roman expected.984 Summer boyc The boyc of the 1984 Summer in Los Angeles years after U.S.-led boycott of the 1980 Summer Olympics in The boycott involved 14 Bloc countries and allies, led by the Soviet Union, which initi boycott on May 8, 1984. Bootting countries organized major event, the Friendship Games, in July and August 1984 Although the boycott by the Soviet Union affected number Olympic Sum were perhaps the first games of a new era to make a profit. Although a boycott led by the Soviet Union depleted the field in certain sports, 140 National Olympic Committees took part, which was a record at the time. Again, without the participation of the Eastern European countries, the 1984 Games were dominated by their host country. The Games were also the first time mainland China (People's Republic) participated. According to British journalist Andrew Jennings, a KGB colonel stated that the agency's officers had posed as anti-doping authorities from	[1] 1984 Summer Olympics boycott The boycott involved 14 Eastern Bloc countries and allies, led by the Soviet Union, which initiated the boycott on May 8, 1984. ... [3] Summer Olympic Games Eastern Bloc that did attend the 1984 Olympics. Although a boycott led by the Soviet Union depleted the field, 140 NOCs took part. Without Eastern European countries, the 1984 Games were dominated by the host. The Games were also the first time mainland China participated. ... [5] 1984 Summer Olympics boycott However, no threat to Eastern Bloc athletes was discovered, and the athletes from the Eastern Bloc country that did attend the 1984 games— Romania —encountered no problems.
Answer	Romania	
Predict	China	Romania

Table 13: Zero-shot QA prompt evaluation of compressor performance across different top-k scenarios, models, and datasets, measured by EM, F1, and inference latency (Lat.). 8B reader experiments were conducted on a single A100-80GB GPU, while 70B reader experiments utilized 4 A100-80GB GPUs in parallel. Best results for each dataset are highlighted in **bold**, Second best results are highlighted in underline.

Compressor	Type	NQ			TQA			HQA			2WIKI			AVG.		
		EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓	EM ↑	F1 ↑	Lat. ↓
Llama-3.1-8B-Instruct																
Top-5 Documents																
Original Docs	-	34.6	47.1	1.0	58.8	68.6	0.9	28.1	38.6	1.0	16.1	24.9	1.1	34.4	44.8	1.0
RECOMP-Abst	Abs.	31.3	43.2	1.6	55.9	65.7	1.4	26.5	37.0	2.2	22.7	29.1	2.1	34.1	43.7	1.8
CompAct	Abs.	32.9	44.6	8.5	58.1	67.7	8.8	28.8	39.8	8.3	16.8	26.0	8.1	34.2	44.5	8.4
Refiner	Abs.	32.9	45.0	28.1	59.2	68.9	10.9	28.8	40.0	6.9	16.8	25.4	6.4	34.4	44.8	13.1
RECOMP-Extr	Ext.	34.6	44.6	**0.5**	56.5	65.1	**0.4**	23.4	32.8	**0.4**	11.2	19.6	**0.6**	31.4	40.5	**0.5**
LongLLMLingua	Ext.	30.2	41.5	0.9	59.4	68.0	0.8	28.0	38	0.8	21.5	27.4	0.9	34.8	43.7	0.9
Ours (EXIT)	Ext.	**35.9**	**47.8**	0.8	**60.8**	**69.9**	0.7	**30.6**	**41.5**	0.8	**24.2**	**30.8**	0.9	**37.9**	**47.5**	0.8
Top-20 Documents																
Original Docs	-	36.6	49.5	3.4	62.0	71.7	2.9	29.9	40.5	2.9	18.8	27.9	3.2	36.8	47.4	3.1
RECOMP-Abst	Abs.	26.9	38.3	1.7	57.3	66.6	1.9	26.8	37.1	2.4	22.7	28.8	2.6	33.4	42.7	2.2
CompAct	Abs.	33.8	45.4	26.1	57.8	67.5	24.5	28.9	39.6	27.5	16.7	24.6	32.2	34.3	44.3	27.6
Refiner	Abs.	30.1	41.4	28.7	57.6	67.0	44.2	26.6	37.3	29.6	16.3	24.9	10.8	32.7	42.6	28.3
RECOMP-Extr	Ext.	32.8	42.6	**0.6**	55.5	63.6	**0.4**	22.2	31.2	**0.5**	10.0	18.3	**0.7**	30.1	38.9	**0.6**
LongLLMLingua	Ext.	33.4	45.1	2.8	62.4	71.2	2.7	31.2	41.4	2.8	24.1	30.1	2.9	37.8	46.9	2.8
Ours (EXIT)	Ext.	**38.1**	**50.8**	1.8	**62.8**	**72.0**	1.7	**32.9**	**44.0**	1.8	**25.5**	**32.3**	2.0	**39.8**	**49.8**	1.8
Llama-3.1-70B-Instruct																
Top-5 Documents																
Original Docs	-	35.6	48.0	8.6	65.1	73.9	7.7	33.7	44.5	8.3	20.8	28.3	9.1	38.8	48.7	8.4
RECOMP-Abst	Abs.	34.1	47.0	4.5	61.3	70.6	3.3	30.3	40.8	4.4	24.2	30.3	4.2	37.5	47.2	4.1
CompAct	Abs.	34.1	45.4	11.9	62.6	71.1	11.7	33.8	44.1	11.0	20.5	27.4	11.6	37.8	47.0	11.5
Refiner	Abs.	35.3	47.1	42.5	64.3	73.0	18.3	33.8	44.7	14.6	21.2	28.0	11.2	38.7	48.2	21.6
RECOMP-Extr	Ext.	35.8	45.3	**2.5**	63.5	71.0	**2.2**	27.6	36.7	**2.9**	13.8	19.3	**3.3**	35.2	43.1	**2.7**
LongLLMLingua	Ext.	32.2	44.0	4.4	66.7	75.2	3.9	34.1	45.3	4.0	28.3	**34.8**	4.3	40.3	49.8	4.1
Ours (EXIT)	Ext.	**36.9**	**49.4**	3.9	**67.3**	**75.9**	3.1	**37.0**	**48.3**	3.3	**28.6**	34.5	3.5	**42.5**	**52.0**	3.5
Top-20 Documents																
Original Docs	-	**39.5**	52.5	25.8	**69.1**	**77.6**	24.9	38.5	50.0	25.3	28.8	**36.8**	28.1	44.0	**54.2**	26
RECOMP-Abst	Abs.	31.5	45.1	4.5	63.4	72.2	3.7	31.3	41.8	4.8	25.4	30.7	4.8	37.9	47.4	4.5
CompAct	Abs.	33.9	45.1	30.8	61.7	70.0	28.1	31.7	40.9	32.0	18.5	23.5	36.5	36.4	44.9	31.9
Refiner	Abs.	32.6	43.5	37.9	62.9	71.4	48.9	31.9	42.2	33.0	22.7	28.7	12.8	37.5	46.5	33.2
RECOMP-Extr	Ext.	33.6	42.7	**2.4**	63.1	70.3	**2.2**	25.6	34.5	**2.9**	12.2	17.4	**3.3**	33.6	41.2	**2.7**
LongLLMLingua	Ext.	34.5	46.4	10.9	68.2	76.9	10.4	36.7	48.4	10.6	29.8	36.5	11.0	42.3	52.1	10.7
Ours (EXIT)	Ext.	39.4	**52.6**	5.1	68.7	77.3	4.3	**38.6**	**50.2**	4.7	**30.0**	36.3	4.8	**44.2**	54.1	4.7

Table 14: Few-shot QA prompt evaluation measured by EM and F1. Best results for each dataset are highlighted in **bold**, Second best results are highlighted in underline.

Compressor	Type	NQ		TQA		HQA		2WIKI		AVG.	
		EM↑	F1↑	EM↑	F1↑	EM↑	F1↑	EM↑	F1↑	EM↑	F1↑
Llama-3.1-8B-Instruct											
Top-5 Documents											
Original Docs	-	**36.9**	**48.8**	**61.5**	**70.3**	29.6	39.9	22.2	29.1	37.5	47.0
RECOMP-Abst	Abs.	33.9	45.1	57.8	66.7	27.0	37.2	**26.2**	**32.2**	36.2	45.3
CompAct	Abs.	35.0	46.3	60.3	69.2	**30.2**	**40.6**	23.9	31.0	37.3	46.8
Refiner	Abs.	34.4	46.0	61.0	70.0	29.6	39.9	23.4	30.0	37.1	46.5
RECOMP-Extr	Ext.	35.9	45.8	58.5	66.1	25.9	35.4	21.2	27.5	35.4	43.7
LongLLMLingua	Ext.	30.7	41.8	60.8	68.8	27.3	37.1	23.0	28.8	35.5	44.1
EXIT (Ours)	Ext.	35.8	47.4	61.0	69.8	29.7	40.3	25.9	32.0	**38.1**	**47.4**
Top-20 Documents											
Original Docs	-	**39.2**	**51.4**	**64.2**	**73.2**	30.6	40.8	24.8	32.4	39.7	49.4
RECOMP-Abst	Abs.	30.2	40.7	59.3	67.6	27.4	37.7	**26.8**	32.4	35.9	44.6
CompAct	Abs.	35.6	47.0	60.1	69.2	30.7	40.6	21.0	28.3	36.9	46.3
Refiner	Abs.	32.2	43.1	59.6	68.6	27.8	37.9	22.8	29.4	35.6	44.7
RECOMP-Extr	Ext.	34.2	43.7	57.5	64.9	24.6	33.8	19.8	26.0	34.0	42.1
LongLLMLingua	Ext.	33.8	45.2	63.8	72.1	31.0	41.1	25.3	31.6	38.5	47.5
EXIT (Ours)	Ext.	38.8	50.8	63.4	72.3	**32.2**	**43.1**	26.6	**32.9**	**40.3**	**49.8**
Llama-3.1-70B-Instruct											
Top-5 Documents											
Original Docs	-	39.5	51.8	68.3	76.5	36.0	46.7	31.4	37.5	43.8	53.1
RECOMP-Abst	Abs.	38.1	50.3	63.4	72.4	30.8	41.2	27.8	33.4	40.0	49.3
CompAct	Abs.	37.9	49.7	67.7	76.0	36.8	47.5	32.2	38.7	43.7	53.0
Refiner	Abs.	38.2	50.2	67.7	76.1	36.0	46.9	30.5	36.6	43.1	52.4
RECOMP-Extr	Ext.	**40.1**	50.9	68.1	75.6	30.8	40.2	26.5	31.9	41.4	49.7
LongLLMLingua	Ext.	35.2	47.2	69.3	77.2	35.6	46.8	34.7	40.2	43.7	52.8
EXIT (Ours)	Ext.	39.5	**51.9**	**69.6**	**77.8**	**38.1**	**49.4**	**35.4**	**41.0**	**45.7**	**55.1**
Top-20 Documents											
Original Docs	-	42.1	55.0	**71.1**	**79.1**	39.4	**51.1**	35.4	**42.4**	47.0	**56.9**
RECOMP-Abst	Abs.	37.2	50.0	65.6	74.0	32.4	43.2	30.3	35.7	41.4	50.8
CompAct	Abs.	37.6	49.1	66.4	74.4	33.6	42.7	25.6	30.5	40.8	49.2
Refiner	Abs.	36.5	47.6	66.6	74.7	33.3	43.3	30.2	35.6	41.6	50.3
RECOMP-Extr	Ext.	38.6	49.1	68.4	75.6	28.9	38.0	24.7	29.9	40.1	48.2
LongLLMLingua	Ext.	37.0	49.1	70.5	78.5	37.9	49.4	35.4	41.1	45.2	54.6
EXIT (Ours)	Ext.	**42.5**	**55.3**	71.0	79.0	**39.8**	**51.1**	**36.5**	42.2	**47.5**	**56.9**

Table 15: Token distribution analysis across different top-k scenarios, models, and datasets. Best results for each dataset are highlighted in **bold**, Second best results are highlighted in underline.

Compressor	Type	NQ	TQA	HQA	2WIKI	AVG.
		#token (%) ↓	#token (%) ↓	#token (%) ↓	#token (%) ↓	#token (%) ↓
Top-5 Documents						
Original Docs	-	723.9 (100.0)	730.3 (100.0)	749.2 (100.0)	784.8 (100.0)	734.4 (100.0)
RECOMP-Abst	Abs.	**38.0 (5.2)**	**36.9 (5.0)**	**63.3 (8.4)**	**55.1 (7.0)**	**46.0 (6.2)**
CompAct	Abs.	77.5 (10.7)	79.0 (10.8)	77.3 (10.3)	71.4 (9.1)	78 (10.6)
Refiner	Abs.	115.6 (16.0)	103.2 (14.1)	<u>76.6 (10.2)</u>	<u>62.1 (7.9)</u>	98.5 (13.4)
RECOMP-Extr	Ext.	<u>43.9 (6.1)</u>	<u>42.7 (5.8)</u>	90.2 (12.0)	97.9 (12.5)	<u>58.9 (8.0)</u>
LongLLMLingua	Ext.	224.3 (31)	221.9 (30.4)	229.2 (30.6)	234.5 (29.9)	225.1 (30.7)
Ours (EXIT)	Ext.	283.8 (39.2)	211.3 (28.9)	190.3 (25.4)	154.9 (19.7)	228.4 (31.2)
Top-20 Documents						
Original Docs	-	2897.4 (100.0)	2925.2 (100.0)	2996.6 (100.0)	3139.7 (100.0)	2939.8 (100.0)
RECOMP-Abst	Abs.	**26.1 (0.9)**	**38.6 (1.3)**	**64.5 (2.2)**	**51.1 (1.6)**	**43.1 (1.5)**
CompAct	Abs.	105.4 (3.6)	102.5 (3.5)	111.8 (3.7)	109.5 (3.5)	106.5 (3.6)
Refiner	Abs.	232.4 (8.0)	176.0 (6.0)	117.3 (3.9)	<u>92.7 (3.0)</u>	175.2 (6.0)
RECOMP-Extr	Ext.	<u>43.4 (1.5)</u>	<u>40.6 (1.4)</u>	<u>89.3 (3.0)</u>	95.7 (3.0)	<u>57.8 (2.0)</u>
LongLLMLingua	Ext.	550.9 (19.0)	553.9 (18.9)	563.3 (18.8)	595.5 (19.0)	556 (18.9)
Ours (EXIT)	Ext.	1001.2 (34.6)	635.0 (21.7)	465.0 (15.5)	367.1 (11.7)	700.4 (23.9)

Table 16: Latency analysis across different top-k scenarios, models and datasets. Each entry shows compression/reading/total time in seconds. 8B reader experiments were conducted on a single A100-80GB GPU, while 70B reader experiments utilized 4 A100-80GB GPUs in parallel. Best results for each dataset are highlighted in **bold**, Second best results are highlighted in underline.

Compressor	Type	NQ			TQA			HQA			2WIKI			AVG.		
		Comp. ↓	Read ↓	Total ↓	Comp. ↓	Read ↓	Total ↓	Comp. ↓	Read ↓	Total ↓	Comp. ↓	Read ↓	Total ↓	Comp. ↓	Read ↓	Total ↓
Llama-3.1-8B-Instruct																
Top-5 Documents																
Original Docs	-	-	1.03	1.03	-	0.93	0.93	-	0.96	0.96	-	1.12	1.12	-	1.03	1.03
RECOMP-Abst	Abs.	1.13	**0.43**	1.55	1.06	**0.29**	1.35	1.92	**0.32**	2.24	1.73	**0.38**	2.11	1.46	**0.36**	1.81
CompAct	Abs.	8.04	**0.43**	8.47	8.47	0.36	8.83	7.80	0.47	8.26	7.65	0.49	8.14	7.99	0.44	8.43
Refiner	Abs.	27.40	0.70	28.10	10.50	0.40	10.90	6.50	0.40	6.90	5.80	0.50	6.40	12.52	0.55	13.07
RECOMP-Extr	Ext.	**0.04**	0.50	**0.54**	**0.04**	0.31	**0.35**	**0.04**	0.37	**0.41**	**0.04**	0.59	**0.63**	**0.04**	0.44	**0.48**
LongLLMLingua	Ext.	0.38	0.47	0.85	0.37	0.42	0.79	0.39	0.46	0.84	0.40	0.53	0.93	0.39	0.47	0.86
Ours (EXIT)	Ext.	0.33	**0.43**	0.76	0.35	0.36	0.71	0.38	0.40	0.78	0.39	0.53	0.92	0.36	0.43	0.79
Top-20 Documents																
Original Docs	-	-	3.41	3.41	-	2.85	2.85	-	2.94	2.94	-	3.22	3.22	-	3.11	3.11
RECOMP-Abst	Abs.	1.07	0.63	1.70	1.55	0.31	1.86	2.09	**0.35**	2.44	2.16	**0.44**	2.60	1.72	**0.43**	2.15
CompAct	Abs.	25.58	0.49	26.06	24.08	0.39	24.47	27.02	0.52	27.53	31.63	0.57	32.19	27.08	0.49	27.57
Refiner	Abs.	28.00	0.70	28.70	43.70	0.50	44.20	29.00	0.60	29.60	9.80	1.00	10.80	27.63	0.69	28.32
RECOMP-Extr	Ext.	**0.11**	0.51	**0.62**	**0.11**	**0.28**	**0.39**	**0.12**	0.42	**0.54**	**0.11**	0.56	**0.68**	**0.11**	0.44	**0.55**
LongLLMLingua	Ext.	1.76	1.04	2.80	1.78	0.92	2.70	1.83	0.93	2.76	1.89	1.01	2.90	1.81	0.98	2.79
Ours (EXIT)	Ext.	1.33	**0.42**	1.76	1.37	0.36	1.74	1.45	0.40	1.85	1.51	0.53	2.04	1.42	**0.43**	1.85
Llama-3.1-70B-Instruct																
Top-5 Documents																
Original Docs	-	-	8.63	8.63	-	7.70	7.70	-	8.30	8.30	-	9.09	9.09	-	8.43	8.43
RECOMP-Abst	Abs.	1.28	3.20	4.48	1.20	**2.14**	3.34	2.06	**2.37**	4.43	1.71	**2.54**	4.24	1.56	**2.56**	4.12
CompAct	Abs.	8.77	3.11	11.88	8.97	2.72	11.69	8.28	2.73	11.01	8.36	3.23	11.59	8.59	2.95	11.54
Refiner	Abs.	35.90	6.60	42.50	14.70	3.60	18.30	11.30	3.30	14.60	8.00	3.20	11.20	17.48	4.16	21.64
RECOMP-Extr	Ext.	**0.04**	**2.44**	**2.48**	**0.04**	2.21	**2.25**	**0.04**	2.87	**2.91**	**0.05**	3.29	**3.33**	**0.04**	2.70	**2.74**
LongLLMLingua	Ext.	0.50	3.87	4.37	0.50	3.40	3.90	0.51	3.52	4.03	0.54	3.72	4.26	0.51	3.63	4.14
Ours (EXIT)	Ext.	0.44	3.50	3.94	0.44	2.66	3.10	0.42	2.88	3.30	0.50	3.03	3.54	0.45	3.02	3.47
Top-20 Documents																
Original Docs	-	-	25.78	25.78	-	24.85	24.85	-	25.35	25.35	-	28.09	28.09	-	26.02	26.02
RECOMP-Abst	Abs.	1.13	3.36	4.49	1.58	2.11	3.70	2.24	2.59	4.83	2.07	**2.75**	4.81	1.76	2.70	4.46
CompAct	Abs.	27.60	3.24	30.84	25.41	2.74	28.15	28.90	3.08	31.99	33.54	2.92	36.45	28.86	2.99	31.86
Refiner	Abs.	32.50	5.40	37.90	47.40	**1.50**	48.90	31.40	**1.50**	33.00	9.70	3.10	12.80	30.25	2.90	33.15
RECOMP-Extr	Ext.	**0.11**	**2.27**	**2.38**	**0.12**	2.10	**2.21**	**0.12**	2.77	**2.89**	**0.13**	3.18	**3.31**	**0.12**	**2.58**	**2.70**
LongLLMLingua	Ext.	2.35	8.51	10.86	2.35	8.04	10.38	2.40	8.24	10.65	2.50	8.47	10.97	2.40	8.32	10.71
Ours (EXIT)	Ext.	1.62	3.50	5.12	1.64	2.67	4.31	1.78	2.88	4.65	1.80	2.96	4.75	1.71	3.00	4.71