# COMPACT: Compressing Retrieved Documents Actively for Question Answering

**Chanwoong Yoon**[1] **Taewhoo Lee**[1] **Hyeon Hwang**[1] **Minbyul Jeong**[1,2†‡] **Jaewoo Kang**[1,3‡]
Korea University[1]
Upstage AI[2]
AIGEN Sciences[3]
{cwyoon99, taewhoo, hyeon-hwang, minbyuljeong, kangj}@korea.ac.kr

## Abstract

Retrieval-augmented generation supports language models to strengthen their factual groundings by providing external contexts. However, language models often face challenges when given extensive information, diminishing their effectiveness in solving questions. Context compression tackles this issue by filtering out irrelevant information, but current methods still struggle in realistic scenarios where crucial information cannot be captured with a single-step approach. To overcome this limitation, we introduce **COMPACT**, a novel framework that employs an active strategy to condense extensive documents without losing key information. Our experiments demonstrate that COMPACT brings significant improvements in both performance and compression rate on multi-hop question-answering benchmarks. COMPACT flexibly operates as a cost-efficient plug-in module with various off-the-shelf retrievers or readers, achieving exceptionally high compression rates (47x).[1]

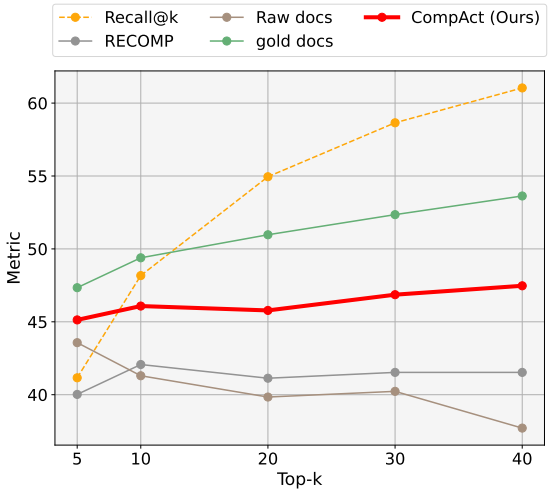


Figure 1: Performance of HotpotQA with different top-$k$ documents, using LLaMA3-8B (Dubey et al., 2024) as the reader. **COMPACT** shows solid performance improvements that align with those of gold documents. This highlights COMPACT's ability to effectively leverage the benefits of increased top-$k$, unlike other methods that struggle with noisy context.

## 1 Introduction

Retrieval-augmented generation empowers language models to solidify their factual grounding, presenting relevant contexts to answer questions (Khandelwal et al., 2019; Lewis et al., 2020; Karpukhin et al., 2020a; Izacard et al., 2023). While these approaches extend the knowledge scope of language models beyond their inherent capabilities, they also introduce challenges when it comes to handling long contexts (Li et al., 2024; An et al., 2024; Qian et al., 2024). First, models often struggle to find key information within extensive contexts, which diminishes their abilities to reference documents (Liu et al., 2024). Also, they often fail to integrate information across multiple documents, which is a common occurrence in real-world scenarios (Cheng et al., 2024). To this end, there is a growing need for methods that can assist models with handling long contexts.

One way to address these challenges is by compressing contexts into concise forms (Li et al., 2023; Pan et al., 2024). The main goal of compression is to reduce the amount of tokens from the original text without losing core information. However, simply compressing contexts can be suboptimal for QA tasks (Joshi et al., 2017; Kwiatkowski et al., 2019), where important details may be filtered out during the compression process (Li et al., 2023). Maintaining redundant information without compression can harm performance, as it may serve as a distractor that can induce models to generate incorrect responses. To handle these limitations, query-focused compression emerges as an effective approach, which aims to preserve information rele-

[†]This work was done while the author was at Korea University.
[‡]Corresponding authors.
[1]Code: https://github.com/dmis-lab/CompAct.



*Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21424–21439
November 12-16, 2024 

vant to the question (Jiang et al., 2023c; Xu et al., 2024; Cao et al., 2024).

However, existing query-focused compressors still struggle to take advantage of information located behind lengthy contexts, leaving out potential opportunities for reader models to improve their answers. In Figure 1, the increase in retrieval recall parallel to the number of documents indicates that useful information is still present even in the lower-ranked results. This demonstrates that these documents should also be covered to fully exploit given information.

Furthermore, existing methods lack the ability to integrate information across multiple documents, which is required in real-world scenarios (Gutiérrez et al., 2024). Figure 2 depicts an example: the question is *"What 'Virtual Choir'-noted conductor has created works for the Austin-based ensemble Conspirare?"*. To answer this, not only do we need to retrieve information implied within the question, but we should also holistically connect and synthesize information across multiple documents. In other words, the quality of answers hinges on the ability of models to dynamically integrate information across multiple documents, which is yet to be fully explored in the field of compression.

To this end, we propose **COMPACT**, a novel framework that can address these challenges by using an active strategy to compress extensive documents and retain crucial information. Our framework has two key components: active compression and early termination. During compression, the model actively encapsulates input documents by jointly analyzing previously compressed contexts with newly provided segments. This ensures that only the most relevant information (here we refer to the compressed text) to the question is preserved at each step, creating a dense and compact context. At each step, the model then decides whether to terminate the compression process. This decision is made based on the relevance and completeness of the information gathered to answer the query.

Our approach offers two distinct advantages. First, it effectively captures essential context from long documents by incorporating segments with the previously compressed context. This is crucial for complex QA tasks that require in-depth reasoning and synthesis of information. Second, it condenses large volumes of documents with a high compression rate, without missing essential contexts. We demonstrate that our framework brings significant improvement in compression rate and end performance in multi-document QA benchmarks. This highlights the effectiveness of our framework, as it preserves the necessary context without losing critical information.

Our contributions are as follows: (1) We propose **COMPACT**, a novel framework that employs an active strategy for compressing extensive documents. We address the challenge of handling long contexts by dynamically preserving query-related contexts, focusing on integrating information across documents. (2) Our framework outperforms existing compressors by a significant margin, achieving a 7.0 (F1) improvement on HotpotQA (Yang et al., 2018) with a higher compression rate (47x). Also, it surpasses the performance of long-context large language models in multi-document QA benchmark datasets. (3) We demonstrate the compatibility of COMPACT with various retrievers and readers, underscoring its effectiveness as a plug-in module between retrievers and readers.

## 2 Preliminaries

### 2.1 Multi-Document Question Answering

Multi-document (or multi-hop) question answering (QA) involves the task of answering questions that require gathering information from multiple documents (Yang et al., 2018; Ho et al., 2020b; Chen et al., 2020; Trivedi et al., 2022; Mavi et al., 2022). This task is more complicated than single-document QA since it requires models to locate and combine information scattered across multiple sources. Even if models can afford lengthy input contexts, they still face challenges in effectively integrating dispersed information from documents.

### 2.2 Multi-hop Information-Seeking

Recent multi-hop information-seeking methods aim to traverse and integrate information across documents by constructing structured maps, such as knowledge graphs or memory trees, over the document context (Wang et al., 2024; Chen et al., 2023; Lee et al., 2024). However, these approaches require an initial building step to create a structured representation of the context. Additionally, they usually navigate their maps to find an optimal traverse path, which demands iterative reasoning by a highly capable model. While we similarly go through the navigation task, we focus on reducing the amount of information the agent has to process, thereby minimizing the computational burden of the reader agent.

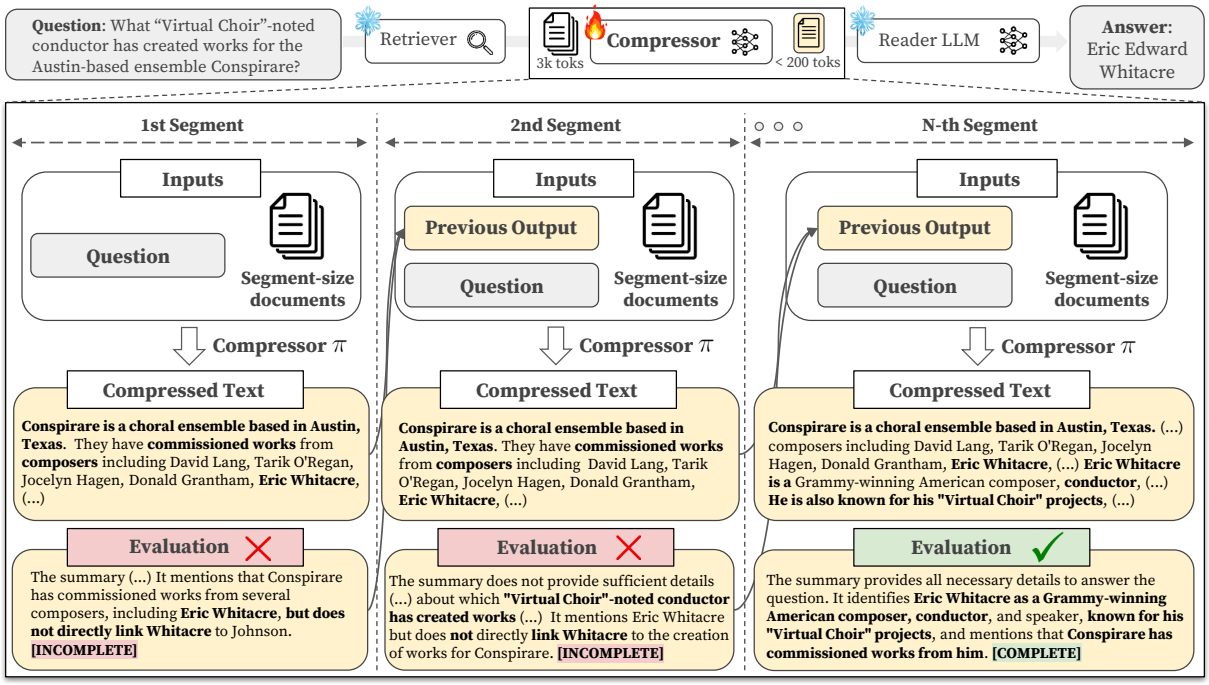


Figure 2: Overall COMPACT framework as a plug-in module between the retriever and the reader LLM. After splitting retrieved documents into segments, COMPACT sequentially compresses these segments into a compacted context. By jointly analyzing the previous context with newly provided segments, we actively compress input documents while preserving essential information in the compressed context. If the segments do not offer complete information to answer the question (1st and 2nd segments), COMPACT continues to the next step to acquire new information. Once all supporting clues are fully captured ($N$-th segment), the iteration ends.

### 2.3 Context Compression

Several studies have focused on compressing the inputs of language models to reduce inference cost while preserving core information. Mu et al. (2024) introduce gisting, a method that compresses input prompts into shorter transformer activations that can be generalized to unseen prompts. ICAE (Ge et al., 2024) proposes training objectives that compress contexts to be restored as closely as possible to the original. Selective-Context (Li et al., 2023) and LLMLingua (Jiang et al., 2023b) utilize conditional probabilities of LLMs to assess the importance of information within contexts. xRAG (Cheng et al., 2024) uses modality fusion to embed document representations into language models and achieves high compression rates.

Additionally, some works have focused on lengthy context inputs. For example, AutoCompressors (Chevalier et al., 2023) transform segments of input context into soft prompts, which are then attached to the next segment as summary vectors. LongLLMLingua (Jiang et al., 2023c) select candidates from documents and then perform token-level compression to retain valuable information relevant to a question. Concurrent with our work, Chain-of-Agents (Zhang et al., 2024) has utilized an iterative framework, which enables information aggregation and context reasoning over long-context tasks. However, our work aims to address a crucial aspect by integrally linking and synthesizing pivotal information between segments while compressing contexts.

### 2.4 Task Formulation

In retrieval-augmented generation, a model $M$ predicts an output $y$ conditioned on an input $x$ and $k$ retrieved passages $D_k = \{d_1, ..., d_k\}_{i=1}^{k}$. For the task of question answering, the input $x$ typically consists of a question $q$ along with an instruction. Thus, $M$ generates an answer $y$ based on $x$ and the retrieved documents $D_k$ as follows: $M(y|x, D_k)$.

To mitigate the costs of $M$ caused by processing a large number of tokens, several approaches have been recently proposed to compress the documents into a shorter context (Wang et al., 2023; Xu et al., 2024). Building on these approaches, our goal is described as follows:

$$arg\max_{\pi} P_M(y \mid C_\pi, x)$$

$$C_\pi = \pi(q, D_k) \quad \text{with} \quad l(C_\pi) \ll l(D_k)$$

where $l$ represents the number of tokens and $\pi$ is a function that compresses documents $D_k$ into a

shorter context $C_\pi$ based on the question $q$. It is important to note that we do not aim to optimize the model $M$ or the retriever. Instead, our primary focus is on compressing the provided contexts into a concise format to ensure that the essential information is retained for answering the question.

## 3 COMPACT

We introduce **COMPACT**, a novel compression framework that actively compresses documents until all necessary evidence for answering a question. To condense a large amount of information from documents, we devise an iterative architecture where the compressed contexts are updated at each iteration. In this section, we provide a comprehensive explanation of our framework and detail the data construction process for training our model.

### 3.1 Active Compression

We reconsider compression as sequential updates of contexts based on the previous information. Figure 2 clearly shows the concept of our framework. Given a question and documents $D_k = \{d_1, ..., d_k\}_{i=1}^{k}$ from a retrieval system, we first group the documents as follows:

$$S_t = \{d_{(t-1)\times j+1}, d_{(t-1)\times j+2}, ..., d_{(t-1)\times j+j}\}$$

where $S_t$ is a $t$-th segment consisting of $j$ documents, and $j$ represents the predefined number of documents to be compressed at each iteration. For example, $S_1 = \{d_1, d_2, ..., d_5\}$ when $j = 5$. We then begin compressing each segment iteratively until it satisfies the end condition. It can be formulated as follows:

$$C_t, E_t = \pi(q, S_t, C_{t-1})$$

Here, $q$ is a given question to answer. $C_t$ and $E_t$ represent the compressed context and an evaluation at step $t$, respectively. $C_t$ is used as part of the input for the next step. During compression, the model actively integrates information related to the question by jointly analyzing the previously compressed context with a newly provided segment. This approach ensures that only the most relevant information is preserved at each step, resulting in a more compact context. As the output context is designed to preserve query-related information, it serves as a comprehensive memory of all iterations up to the current step. We describe an example in Table 13.

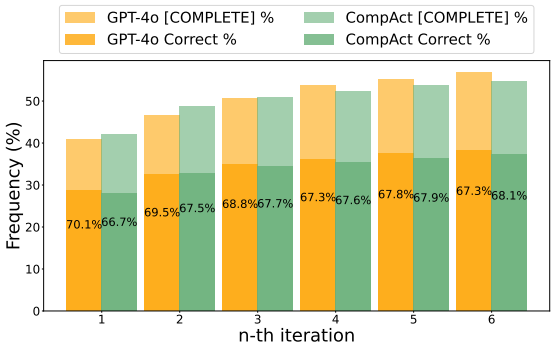


Figure 3: Distribution of iteration points where models determine the compressed contexts to be complete. The frequencies of completeness are accumulated over iterations. We compare the distribution between GPT-4o (Yellow) and COMPACT (Green). We also measure the percentage of correctness at each iteration, using an F1 score of 0.4 as the threshold for correction.

### 3.2 Early Termination

To ensure that the iteration does not continue unnecessarily once enough information is obtained, we introduce a specific end condition for early termination. We implement this by including an evaluation $E$ in the generation process to decide the endpoint. The evaluation $E$ consists of a rationale and a condition token ([COMPLETE] or [INCOMPLETE]). The purpose of $E$ is to assess whether an input segment $S_t$, combined with the previous context $C_{t-1}$, provides sufficient details to answer the question. If the token indicates that the provided context is sufficient, the iteration terminates; otherwise, we continue to gather missing information until all details are fully obtained.

This early termination offers three primary benefits. First, it prevents redundant contexts from entering the compressed contexts or acting as a distraction. Second, it avoids meaningless iterations, thereby drastically lowering the computational burden that may stem from iterative processing steps. Third, it dynamically adjusts to the complexity of the question and the information density of the documents. This flexibility enables our COMPACT framework to be both effective and efficient across a wide range of scenarios, from simple questions to more complex, multi-hop questions that require extensive context integration.

### 3.3 Dataset Construction

We compress documents into a query-related context while concurrently determining the endpoint of the iterations. To cultivate this capability, we instruct a superior LLM to follow a three-step pro-

| Dataset | [COMPLETE] | | [INCOMPLETE] | | Total |
|---|---|---|---|---|---|
| | first | subsequent | first | subsequent | |
| HotpotQA | 7.2K | 7.2K | 7.2K | 7.2K | 28.8K |

Table 1: Statistics of our generated dataset. We categorize it into four cases: [COMPLETE] and [INCOMPLTE], each further split based on whether it is the first or subsequent iteration.

cess. We provide the prompt in Table 12.

**Sentence-Level Selection.** We begin by asking the LLM to analyze sentences, particularly focusing on relevant clues that may help answer the question. If certain sentences provide relevant information or implicitly clarify ambiguous points within the question, the LLM is instructed to generate these sentences from the provided documents.

**Query-focused Compression.** We then generate a compressed text of the selected sentences based on the question. We explicitly restrict the LLM from making assumptions or attempting to conclude without supporting evidence, as follows: *"DO NOT make assumptions or attempt to answer the question; your job is to summarize only."*. This restriction is crucial because our main objective here is to condense relevant information from the provided documents, instead of directly answering the questions. Skipping the logical steps required to answer the question, as if relying on parametric knowledge, can harm compression performance by increasing the likelihood of missing essential information.

**Determining the Early Termination.** We also prompt the LLM to evaluate its own compressed contexts based solely on the provided information, without any additional background context. We direct the LLM to generate a condition token (e.g., [COMPLETE] or [INCOMPLETE]) along with the rationale for its judgment.

Overall, we construct a synthetic dataset by instructing the LLM based on the three-step processes described above. Table 1 shows the dataset statistics. We conduct data construction from two scenarios: realistic and distractor. In realistic scenarios, provided documents are the results of a retrieval system. However, due to the retriever's limited performance, gold documents rarely appear, which can hinder the collection of cases with early termination. This results in a scarcity of cases in the dataset where the iteration is terminated early (i.e. [COMPLETE] at a subsequent iteration). To address this issue, we collect data from distractor scenarios which include predefined documents that contain all supporting facts needed to answer the question. After filtering the collected datasets from both scenarios, we build a training dataset consisting of 28k instances categorized into four distinct groups.

## 4 Experiment

### 4.1 Experimental Setup

**Dataset Construction** We use the GPT-4o (OpenAI, 2024) API (2024-05-13) as the LLM to collect our dataset. We use only a subset of HotpotQA (Yang et al., 2018) training set for data collection. To retrieve documents, we use Contriever (Izacard et al., 2022), fine-tuned on MS-MARCO (Bajaj et al., 2016), as our retrieval system on the 2018 Wikipedia corpus (Karpukhin et al., 2020b). We set the default number of documents per segment $j$ to 5 and top-$k$ to 30, allowing for a maximum of 6 iterations per query. To prevent lengthy API responses, the maximum number of generated tokens is limited to 700.

**Training & Inference** We perform supervised fine-tuning to train our model using the collected dataset. Without using specific labels or methods for particular iterations, we focus on training the model to effectively update the previous context based on the question and given documents at each step. We use instruction-tuned Mistral-7B (Jiang et al., 2023a) as our backbone base model. At inference, we process the same number of segments and inputs as training. Further information is provided in the Appendix D.1.

### 4.2 Datasets

We evaluate COMPACT on both single-document and multi-document question-answering (QA) datasets. For single-document QA, we use Natural Question (NQ) (Kwiatkowski et al., 2019) and TriviaQA (TQA) (Joshi et al., 2017). For multi-document QA, we evaluate on HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultiHopQA (Ho et al., 2020a). The evaluation is conducted on the dev set of each dataset, except for TriviaQA, which is evaluated on the test set. As mentioned, we construct the training data using only a subset of HotpotQA. Therefore, we conducted zero-shot evaluation on the other datasets without accessing their training set.

| Methods | HotpotQA | | | MuSiQue | | | 2WikiMQA | | | NQ | | | TriviaQA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Comp. | EM | F1 | Comp. | EM | F1 | Comp. | EM | F1 | Comp. | EM | F1 | Comp. | EM | F1 |
| Oracle | 10.8x | 39.9 | 51.2 | 10.3x | 14.2 | 23.6 | 11.0x | 37.4 | 43.2 | - | - | - | - | - | - |
| Raw Document | 1x | 29.4 | 40.3 | 1x | 6.5 | 15.6 | 1x | 25.4 | 31.2 | 1x | 39.0 | 51.3 | 1x | 68.9 | 77.1 |
| *Long-Context LLM* | | | | | | | | | | | | | | | |
| InternLM2-chat-7B | 1x | 8.0 | 20.3 | 1x | 1.0 | 6.8 | 1x | 9.3 | 19.5 | 1x | 7.6 | 22.6 | 1x | 12.1 | 31.5 |
| Mistral-7B-Instruct-v0.2 | 1x | 9.5 | 22.6 | 1x | 1.0 | 7.9 | 1x | 1.2 | 15.4 | 1x | 4.3 | 20.9 | 1x | 35.3 | 50.4 |
| FILM-7B | 1x | 32.4 | 43.7 | 1x | 6.9 | 15.7 | 1x | 26.4 | 31.7 | 1x | 38.2 | 50.8 | 1x | 62.7 | 71.7 |
| Phi-3-medium-128k-instruct | 1x | 22.3 | 34.7 | 1x | 5.8 | 14.6 | 1x | 24.8 | 31.0 | 1x | 29.1 | 42.2 | 1x | 61.0 | 70.6 |
| Yi-9B-200k | 1x | 28.6 | 39.4 | 1x | 6.8 | 15.1 | 1x | 25.0 | 30.4 | 1x | 33.9 | 45.2 | 1x | 62.8 | 71.3 |
| Phi-3.5-mini-instruct | 1x | 21.6 | 33.0 | 1x | 4.8 | 12.7 | 1x | 21.0 | 26.8 | 1x | 29.3 | 41.2 | 1x | 58.2 | 67.9 |
| Llama-3.1-8B-Instruct | 1x | 31.4 | 42.9 | 1x | 6.2 | 14.6 | 1x | 30.2 | 36.0 | 1x | 35.8 | 49.3 | 1x | 63.6 | 74.2 |
| GPT-3.5-turbo | 1x | 32.8 | 43.8 | 1x | 7.3 | 16.1 | 1x | 28.6 | 33.9 | 1x | **40.8** | **54.6** | 1x | **69.9** | **77.4** |
| *Compressor* | | | | | | | | | | | | | | | |
| AutoCompressors | 35.4x | 18.4 | 28.4 | 34.7x | 3.9 | 11.9 | 36.2x | 19.0 | 24.5 | 34.4x | 17.3 | 31.8 | 34.5x | 55.3 | 64.3 |
| LongLLMLingua | 3.4x | 25.6 | 35.3 | 3.4x | 4.8 | 13.5 | 3.6x | 27.9 | 32.9 | 3.5x | 27.7 | 40.6 | 3.3x | 64.0 | 70.8 |
| RECOMP (extractive) | 34.3x | 29.7 | 39.9 | 32.7x | 6.7 | 15.7 | 35.9x | 29.9 | 34.9 | 32.7x | 34.6 | 45.1 | 39.2x | 67.6 | 74.1 |
| COMPACT (Ours) | **47.6x** | **35.5** | **46.9** | **37.2x** | **8.7** | **18.1** | **51.2x** | **31.0** | **37.1** | **48.5x** | 38.4 | 50.0 | **49.4x** | 65.4 | 74.9 |

Table 2: Main results. We set the reader as LLaMA3-8b (Dubey et al., 2024). We retrieve top-30 documents. We use three multi-document (HotpotQA, MuSiQue, and 2WikiMQA) and two single-document (NQ and TriviaQA) question-answering datasets. Since our training datasets consist of a subset of HotpotQA, we perform zero-shot evaluation on the rest of the datasets. Comp. refers to the compression rate which is denoted as follows: compression rate $= \frac{\text{\# of tokens in retrieved documents}}{\text{\# of tokens in compressed text}}$.

### 4.3 Baselines

In Table 2, we compare COMPACT against several baseline methods. To ensure a fair comparison, we feed compressed contexts from each baseline to the same reader model, LLaMA3-8B (Dubey et al., 2024). We consider the following baselines. (1) *Oracle.* We provide the reader with documents that contain the answers to the questions. If such documents are not available, we include five documents as a default. (2) *Raw Document.* We simply concatenate the top-$k$ retrieved documents. (3) *Long-Context LLM.* As these LLMs are designed to handle large inputs, they align with our objective of managing extensive contexts, making them suitable for our baselines. We use InternLM2-chat-7B (Cai et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023a), FILM-7B (An et al., 2024), Phi-3-medium-128k-instruct and Phi-3.5-mini-instruct (Abdin et al., 2024), Yi-9B-200k (AI et al., 2024), Llama-3.1-8B-instruct (Dubey et al., 2024), and GPT-3.5-turbo-0125 (OpenAI, 2023). (4) *Compressor.* We compare COMPACT with three compression-based methods: AutoCompressors (Chevalier et al., 2023), RECOMP (Xu et al., 2024), and LongLLMLingua (Jiang et al., 2023c). We provide the detailed descriptions of the baselines in Appendix D.2.

### 4.4 Results

We assess the performance of COMPACT using three metrics: Compression rate (Comp.), Exact Match (EM), and F1 score. Overall, COMPACT exhibits strong performance across all QA benchmark datasets, achieving the highest compression rate among all baselines. Specifically, it surpasses other compression-based methods in all three metrics, demonstrating its strong ability to compress abundant information (∼3k tokens) efficiently.

COMPACT falls short of the performance of GPT-3.5-turbo in single-document QA (NQ and TriviaQA), which may be due to our model being trained exclusively on a subset of HotpotQA. Even with this constraint, our framework outperforms existing compressors and achieves competitive performance with long-context LLMs. Plus, it represents entire contexts using significantly fewer tokens, highlighting its efficiency in providing compact representations. Moreover, in multi-document QA, COMPACT achieves superior performance compared to other baselines. This underscores the persistent challenge of integrating information across multiple documents and emphasizes how COMPACT excels at such tasks.

## 5 Analysis

We investigate ways to facilitate the use of COMPACT as a plug-in module that collaborates with diverse retrievers and readers (Section 5.1). We conduct an ablation study to assess the impact of components on performance (Section 5.2) and examine the cost-effectiveness of our framework using proprietary black-box models (Section 5.3). Fi-

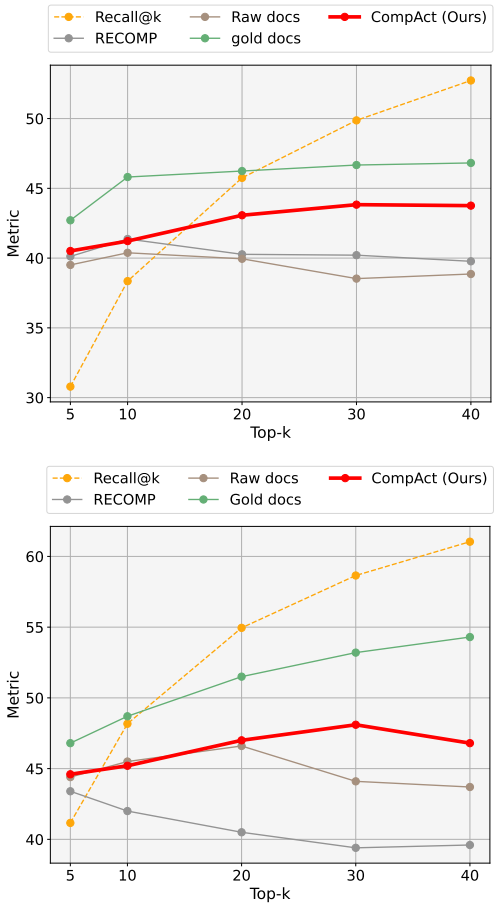


Figure 4: Performance of HotpotQA with different top-$k$ documents, using Contriever (upper) as the retriever and GPT-3.5-Turbo (lower) as the reader.

nally, we discuss computational efficiency involved in our framework (Section 5.4).

### 5.1 Compressor as a Plug-in Module

In Figure 2, we illustrate the compressor as a plug-in module, highlighting that retrievers and readers can be easily replaced by other models. We investigate if CompAct can flexibly compress context provided by diverse retrievers, while preserving useful information regardless of various readers.

**Generalizability across Retrievers.** In Figure 4 and 5, we use Contriever (Izacard et al., 2022) and BM25 (Robertson et al., 2009), two of the most well-known retrievers, to replace source documents. We evaluate our framework with 500 random samples from the HotpotQA (Yang et al., 2018) dev set, using different top-$k$. We compare our results with several baselines: gold documents (oracle), raw documents, and RECOMP (Xu et al., 2024).

With the Contriever setup, where the retriever often fails to locate relevant documents at high-ranking positions, increasing the top-$k$ leads to more distinct performance improvements. This shows that our framework effectively captures and utilizes valuable information from lower-ranked documents. Additionally, in the BM25 setup, CompAct shows consistent performance while retrieving up to top-40 documents. Notably, our framework achieves a similar saturated performance trend to the gold documents setup, indicating its competence in filtering noisy contexts. In both setups, CompAct achieves significantly higher performance compared to other baselines. As we intended, these observations demonstrate that CompAct shows robustness across various retriever setups.

| Components | HotpotQA | | MuSiQue | | 2WikiMQA | |
|---|---|---|---|---|---|---|
| | Comp. | F1 | Comp. | F1 | Comp. | F1 |
| LLaMA3-8B | | | | | | |
| Rationale. | 130.8x | 41.6 | 120.0x | 15.9 | 141.3x | 32.3 |
| CT | 47.5x | **48.3** | 36.5x | **19.1** | 52.2x | **36.2** |
| CT + Rationale | 33.6x | 47.3 | 27.1x | 19.0 | 36.4x | 35.6 |
| LLaMA2-13B | | | | | | |
| Rationale. | 141.8x | 41.8 | 129.2x | 16.9 | 152.4x | 30.8 |
| CT | 48.1x | **48.5** | 37.0x | **18.6** | 52.7x | **35.6** |
| CT + Rationale. | 34.6x | 47.3 | 28.0x | **18.6** | 37.4x | 34.2 |
| GPT-3.5-Turbo | | | | | | |
| Rationale. | 135.2x | 38.0 | 123.5x | 13.8 | 146.2x | 24.0 |
| CT | 48.1x | **49.2** | 37.0x | **20.9** | 53.0x | 34.0 |
| CT + Rationale. | 33.9x | 47.0 | 27.4x | 18.5 | 36.7x | **36.5** |

Table 3: Results of each component effectiveness. CT refers to the compressed text.

**Generalizability across Readers.** We look into whether CompAct truly provides generalized compressed texts suitable for diverse readers. To this end, we assess the quality of our compressed texts using diverse reader LLMs: GPT-3.5-Turbo (OpenAI, 2023), LLaMA2-13B (Touvron et al., 2023), and LLaMA3-8b (Dubey et al., 2024). Figure 4 presents the results of using GPT-3.5-Turbo as a reader, while figure 6 includes the results for LLaMA2-13B and LLaMA3-8B.

Our results show that CompAct sufficiently delivers high-quality compressed texts applicable to different readers. Also, we prove its effectiveness on the top-$k$ documents with high $k$. In Figure 4, there is little difference in performance up to the top-20 between the raw documents setup and ours. We hypothesize this is attributed to the strong performance of the reader, GPT-3.5-Turbo, in processing moderate length of contexts. However, at the top-30 and top-40 documents, performance degra-

| Model | Raw | | RECOMP | | LINGUA* | | COMPACT | |
|---|---|---|---|---|---|---|---|---|
| | Cost | F1 | Cost | F1 | Cost | F1 | Cost | F1 |
| GPT-3.5-Turbo | 1.09 | 44.5 | **0.04** | 40.1 | 0.33 | 38.4 | **0.04** | **49.2** |
| GPT-4o | 10.75 | 55.8 | 0.43 | 48.1 | 3.31 | 47.6 | **0.28** | **56.0** |
| Claude-3.5 | 6.45 | 36.0 | 0.26 | 37.0 | 1.99 | 30.2 | **0.17** | **42.2** |
| Gemini-1.5-pro | 7.54 | **52.0** | 0.31 | 41.7 | 2.36 | 40.1 | **0.20** | 44.8 |

Table 4: API cost of 500 samples from a HotpotQA dev set. LINGUA* refers to LongLLMLingua. We assess the inference cost (USD) of each method when employing proprietary models as readers.

dation occurs as more documents are included, reflecting the difficulty of handling lengthy documents with increased noisy information. In contrast, COMPACT exhibits marginal performance degradation even with a higher number of documents.

Furthermore, COMPACT achieves a high compression rate above 40x, which significantly reduces the number of input tokens, making it highly cost-effective for API operations. This efficiency, combined with its ability to maintain performance across diverse readers, underscores the superior capability of COMPACT.

### 5.2 Component Effectiveness

COMPACT actively compresses source documents by generating an intermediate compressed text (CT) with termination evaluation for each iteration. The evaluation consists of two components: a rationale explaining the reasons for termination and a condition token to decide the termination. To understand how each component affects end performance, we conduct an ablation study of components as shown in Table 3. we use 500 random samples from each dataset. When only the rationale is provided, the compression rate increases dramatically, but the end performance (EM & F1) significantly drops (Row 1). Conversely, when we only provide compressed text, we achieve the highest performance with most readers. However, when adding the rationale with the compressed text (CT + Rationale), there are no clear benefits; in most cases, performance declines. We hypothesize that some judgments in the rationale distract the readers from generating an answer purely from the compressed context. This could act as a negative shortcut in the answering process, resulting in decreased performance.

### 5.3 Cost Efficiency

To evaluate the cost-saving benefits, we employ four proprietary models as readers: GPT-3.5-Turbo (OpenAI, 2023), GPT-4o (OpenAI, 2024), Claude-3.5-sonnet (Anthropic, 2024), and Gemini-1.5-pro (Google, 2024). In Table 4, we show that our framework achieves superior performance at the lowest cost compared to other baselines. Surprisingly, COMPACT achieves competitive performance to the raw document setups with superior models known to possess exceptional long-context understanding ability. This indicates COMPACT's high-level expertise in compressing contexts.

| Dataset | TFLOPs | | F1 | |
|---|---|---|---|---|
| | Raw | CompAct | Raw | CompAct |
| HotpotQA | **34.1** | 35.8 | 40.0 | **48.3** |
| MusiQue | **33.6** | 49.3 | 16.2 | **19.0** |
| 2WikiMQA | **35.9** | 42.4 | 29.5 | **37.2** |
| NQ | 32.9 | **26.7** | 52.9 | **53.8** |
| TQA | 33.5 | **24.6** | **78.5** | 77.3 |

Table 5: Average TeraFLOPs (TFLOPs) and F1 scores. TFLOPs are normalized by the number of instances. We utilize LLaMA3-8B as a reader with top-30 documents and employ DeepSpeed FlopsProfiler (Rasley et al., 2020) for measurement.

### 5.4 Computational Efficiency

While COMPACT offers a significant cost-saving advantage, we also consider a potential increase in computation due to the active strategy employed by our framework. To assess this, we measure the Floating Point Operations per Second (FLOPs) of our framework in comparison to the baseline raw context setup, as shown in Table 5.

We reveal that COMPACT consistently demonstrates higher performance than the raw context setup in multi-hop QA tasks (HotpotQA, MusiQue, 2WikiMQA). Specifically, on HotpotQA, it achieves a large increase in F1 score while maintaining comparable computational costs. Although MusiQue and 2WikiMQA exhibit higher costs-primarily due to the increased iterations required to identify supporting documents in low recall scenarios-the performance gains are substantial. Conversely, for single-hop QA tasks (NQ, TQA), our framework achieves competitive performance with significantly reduced costs, demonstrating the saving effect of early termination. This highlights how the active strategy allows us to dynamically adjust the computational costs allocated to each instance, making our framework flexible for questions with diverse levels of complexity and varying information requirements.

## 6 Conclusion

We introduce **COMPACT**, a novel framework that employs an active strategy to compress extensive retrieved documents. Our framework effectively captures pivotal information from a large number of documents by dynamically retaining essential contexts and incorporating information. We demonstrate that COMPACT significantly outperforms existing compressors, showing a large performance gap with a higher compression rate in multi-document question-answering benchmarks. Furthermore, it serves as a convenient plug-in module that can seamlessly collaborate with various off-the-shelf retrievers and readers while providing cost-saving benefits.

## Limitations

We acknowledge that COMPACT has a longer inference time when processing retrieved documents, compared to other compressors. Given that our framework contributes to addressing complex question types, which is pioneering in the field of compression, we believe that future research can build upon COMPACT to further improve these issues.

Additionally, even a strong proprietary model like GPT-4o can make mistakes when determining the completeness of given contexts. There may still be error cases in our data construction process, although we attempt to address this issue by filtering them out.

Lastly, we only use Mistral-7B-Instruct-v0.2 as our base model due to resource limitations. Verifying whether COMPACT works well across a range of model sizes, both smaller (< 7B) and larger (> 7B), could lead to interesting findings.

## Ethics Statement

Our training process can incur significant environmental costs due to its computationally intensive nature. To mitigate this, we fine-tune a single Mistral model to minimize computational expenses. Furthermore, a potential risk of this work is that the generated dataset may contain biases from API calls, such as stereotypes related to race and gender. To our knowledge, there haven't been significant issues reported when creating question-answering datasets. However, it would be beneficial to apply methods that robustly train or validate against such concerns.


## Acknowledgments

This work was supported in part by the National Research Foundation of Korea [NRF2023R1A2C3004176], the Ministry of Health & Welfare, Republic of Korea [HR20C002103], the Ministry of Science and ICT (MSIT) [RS-2023-00220195], the ICT Creative Consilience program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the MSIT [IITP-2024-2020-0-01819], and the National Research Foundation(NRF), Korea, under project BK21 FOUR.


## References


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*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. 2024. Make your llm fully utilize the context. *arXiv preprint arXiv:2404.16811*.

Anthropic. 2024. claude-3.5-sonnet.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. 2024. InternLM2 technical report. *arXiv preprint arXiv:2403.17297*.

Zhiwei Cao, Qian Cao, Yu Lu, Ningxin Peng, Luyang Huang, Shanbo Cheng, and Jinsong Su. 2024. Retaining key information under high compression ratios: Query-guided compressor for llms. *arXiv preprint arXiv:2406.02376*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

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. *arXiv preprint arXiv: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*.

Tao Ge, Hu Jing, 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*.

Google. 2024. gemini-1.5-pro.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *arXiv preprint arXiv:2405.14831*.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020b. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on 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. *Transactions on Machine Learning Research*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023c. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. *arXiv preprint arXiv:2310.06839*.

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)*. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020a. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

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

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

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*, 7:453–466.

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*.

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*, 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*.

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

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

OpenAI. 2023. Chatgpt.

OpenAI. 2024. Gpt-4o.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. *arXiv preprint arXiv:2403.12968*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Yujia Zhou, Xu Chen, and Zhicheng Dou. 2024. Are long-llms a necessity for long-context tasks? *arXiv preprint arXiv:2405.15318*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*.

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*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. 2023. The alignment handbook. https://github.com/huggingface/alignment-handbook.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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 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*.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Preprint*, arXiv:2406.02818.

Jared Rasley, Samyam Rajbhandari, Oshrat Ruwase, and Yuxiong He. 2020. DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 3505–3506. https://doi.org/10.1145/3394486.3406703.

Mohamed Abdin, Sebastian Jacobs, Adeel Awan, Jatin Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Mohit Bahree, Ahmad Bakhtiari, Harsh Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

01.AI, Andrew Young, Bin Chen, Chao Li, Chen Huang, Guodong Zhang, Guang Zhang, Haotian Li, Jiaming Zhu, Jin Chen, Jiawei Chang, Kaifeng Yu, Pengfei Liu, Qi Liu, Shang Yue, Shuai Yang, Shuo Yang, Tao Yu, Wei Xie, Wei Huang, Xiaoyi Hu, Xudong Ren, Xinting Niu, Ping Nie, Yihan Xu, Yufei Liu, Yida Wang, Yuxin Cai, Zheng Gu, Zhenghao Liu, and Zhilin Dai. 2024. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2404.14219*.

Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, number 17, pages 19206–19214.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2023. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. 2024. A human-inspired reading agent with gist memory of very long contexts. In *Proceedings of the Forty-first International Conference on Machine Learning*.

## A Practicality of Compressing Contexts

To ensure the practicality of providing context with fewer tokens, we present an additional point to reinforce the necessity of our research. In table 6, we investigate the maximum input length of language models with over 1 million downloads on Huggingface[2]. We find that 77.5% of these models can only afford inputs of 512 tokens or fewer. Despite ongoing research trends on LLMs capable of handling long texts, it is evident that many users still frequently employ models with smaller token inputs. Considering the current state, COMPACT offers substantial benefits to models with smaller input lengths by allowing them to access more information, effectively acting as a bridge.

| Sequence Length | Language Models (%) |
|---|---|
| 128 | 14.7 |
| 512 | 62.8 |
| $\geq$ 1024 | 22.5 |

Table 6: Huggingface Models Statistics. 77.5% of models cannot receive at least top-5 documents as input. We select frequently-used models downloaded at least 1M in https://huggingface.co/Models.

## B Additional Comparison

To further evaluate against additional compression methods, we conduct supplementary experiments following the setup from Cao et al. (2024). The results show that our framework outperforms existing methods on both TriviaQA and HotpotQA, while slightly underperforming on NQ. However, we would like to highlight three key factors that can influence the performance: (1) Training Data: We use only a subset of the HotpotQA training data, resulting in a zero-shot evaluation on the other datasets (NQ, TQA) for evaluating transfer capabilities. (2) Model Weights: Unlike Cao et al. (2024), we do not fine-tune separate model weights for each dataset, demonstrating the versatility of our approach. (3) Reranker Dependency: While Cao et al. (2024) relies on a strong reranker (e.g., Cond.PPL (Jiang et al., 2023c)) to refine the input context, our approach does not rely on external rerankers, instead independently determining the priority of information within the context.

[2]https://huggingface.co/Models

Notably, in HotpotQA, our framework surpasses the oracle setup reported in Cao et al. (2024) where all gold documents are provided. Based on this, we hypothesize that providing complete gold evidence does not always create an optimal context for the reader model. Despite having the necessary information in the oracle, the F1 score remains significantly lower (57.7), indicating that the quality of summaries is critical. Our summaries, which prioritize essential information needed to answer the question, ensure that the context given to the reader model is both relevant and concise, leading to superior performance.

| Methods | NQ | | TQA | | HotpotQA | |
|---|---|---|---|---|---|---|
| | Comp. | Acc | Comp. | EM | Comp. | F1 |
| Oracle | 59.2x | 73.5 | - | - | 42.2x | 57.7 |
| ICAE [2] | **21.5x** | 53.3 | 10.2x | 48.9 | 9.5x | 34.5 |
| QGC [1] | 15.2x | **60.9** | 7.9x | 57.5 | 8.8x | 51.6 |
| ($\epsilon = 0.42$) | 20.6x | 57.6 | **10.9x** | 57.1 | 12.1x | 51.2 |
| CompAct (Ours) | 14.6x | 57.0 | **10.9x** | **64.4** | **12.2x** | **59.0** |

Table 7: Comparison with ICAE (Ge et al., 2024) and QGC (Cao et al., 2024). Following Cao et al. (2024), we use LLaMA2-7B as readers and report the Compression rate (Comp.) and F1 score.

## C Length of Compressed Text

**Token Length of Compressed Text per Iteration** In Table 8, we provide detailed length information of compressed texts per iteration. As the token length slightly increases with each iterations, We observe that COMPACT maintains a high compression rate on average, which compresses 30 documents into under 200 tokens.

| Datasets | N-th Iterations | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| HotpotQA | 78.1 | 114.1 | 128.5 | 126.5 | 135.9 | 147.5 |
| MuSiQue | 77.5 | 110.6 | 135.2 | 91.6 | 145.6 | 124.0 |

Table 8: Average token length of compressed texts per iteration. 5 documents are compressed for each iteration, as default setup of our framework.

**Token Usage** In Table 9, we compare token usage against the baseline raw context setup with separate steps: Compress and Read. While COMPACT generates more output tokens overall, it maintains a similar level of computational cost (see Table 5). This is due to context segmentation, which mitigates the quadratic increase in computational cost

associated with sequence length, resulting in significant computational efficiency benefits.

| Dataset | Raw | CompAct | | |
|---|---|---|---|---|
| | Total | Compress | Read | Total |
| HotpotQA | 5218 / 7 | 4068 / 649 | 347 / 5 | 4415 / 654 |
| MusiQue | 5125 / 8 | 5623 / 963 | 384 / 7 | 6007 / 970 |
| 2WikiMQA | 5453 / 8 | 4927 / 718 | 348 / 7 | 5275 / 724 |
| NQ | 5023 / 8 | 2993 / 503 | 279 / 8 | 3273 / 511 |
| TQA | 5091 / 5 | 2707 / 443 | 342 / 5 | 3049 / 447 |

Table 9: Average token usage (input/output) per instance.

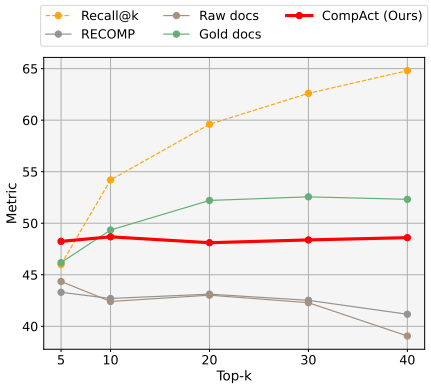


Figure 5: Performance of HotpotQA with different top-$k$ documents, using BM25 as the retriever.

## D Implementation Details

### D.1 Training & Inference

We use 4 Nvidia A100 with 80GB memory to train our CompAct framework. Our code is written in PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2019). We use supervised fine-tuning through published alignment-handbook (Tunstall et al., 2023). We train the model with Adam optimizer (Kingma and Ba, 2015), using a learning rate of 2e-6, a batch size of 64, and 0.1 warm up ratio for 7 epochs. For inference, all experiments are conducted using a greedy decoding strategy with a temperature of 0 and top_p set to 1.0.

### D.2 Baselines

**Long-context LLMs.** InternLM2-chat-7B (Cai et al., 2024) has shown near-perfect performance on the Needle-in-the-Haystack task, which tests how well a model utilizes information within a long context. Mistral-7B-Instruct-v0.2 (Jiang et al., 2023a) has recently shown strong performance across various benchmarks and supports a 32k context window. FILM-7B (An et al., 2024), trained with a synthetic long-context question-answering dataset, has shown strong performance on tasks that require information awareness in the long context. Phi-3-medium-128k-instruct and Phi-3.5-mini-instruct (Abdin et al., 2024), leveraging their custom datasets, achieve state-of-the-art performance with a focus on high-quality reasoning. Yi-9B-200k (AI et al., 2024) is an extended context version of the Yi series models. Llama-3.1-8B-Instruct (Dubey et al., 2024), one of the latest models in the Llama family, supports a 128k context window, enabling enhanced performance on long-context tasks. We also experiment with GPT-3.5-turbo, a popular proprietary LLM that supports a 16k context window.

**Compressors.** (5) AutoCompressors (Chevalier et al., 2023) process segments of long context into soft prompts, which are prepended to the next segment as summary vectors. We use 50 summary tokens for every 2,048 tokens, following the setup from the original paper. (6) LongLLMLingua (Jiang et al., 2023c) takes a perplexity-based approach to filter out tokens with less importance. (7) RECOMP (Xu et al., 2024) suggests an extractive compressor that extracts relevant sentences using a dual encoder model, and an abstractive compressor that summarizes documents using an encoder-decoder model. We experiment with the extractive compressor setting, selecting 4 sentences from documents to ensure a fair comparison at similar text lengths.

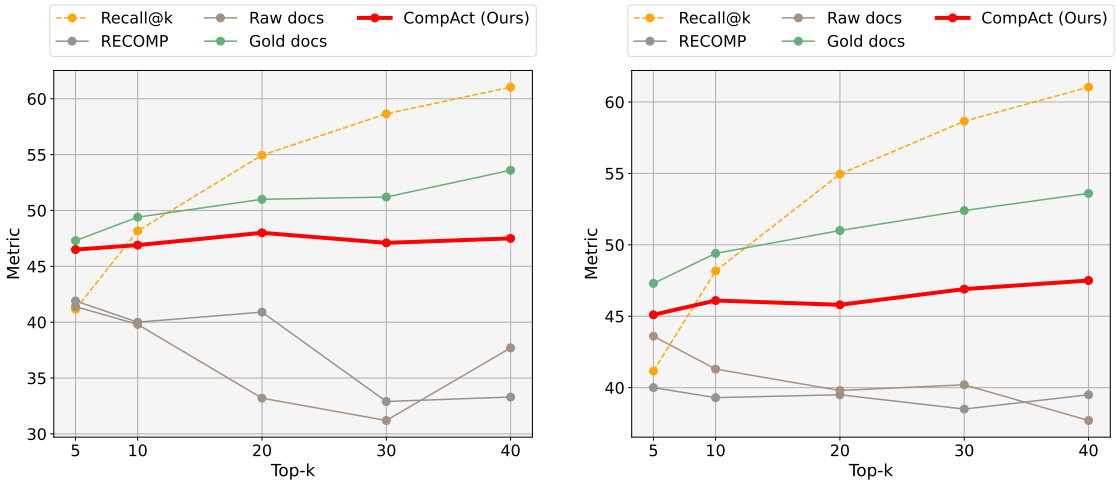


Figure 6: Performance of HotpotQA with different readers: LLaMA2-13B (left) and LLaMA3-8B (right).

| **Dataset** | **Train** | **Dev** | **Test** | **Avg. # of Supporting Documents** | **# of Pre-defined Context** |
|---|---|---|---|---|---|
| NaturalQuestions (Kwiatkowski et al., 2019) | 79,168 | 8,757 | 3,610 | - | - |
| TriviaQA (Joshi et al., 2017) | 78,785 | 8,837 | 11,313 | - | - |
| HotpotQA (Yang et al., 2018) | 90,447 | 7,405 | - | 2 | 10 |
| MuSiQue (Trivedi et al., 2022) | 39,876 | 4,834 | 4,918 | 1.89 (Dev) | 20 |
| 2WikiMultiHopQA (Ho et al., 2020a) | 167,454 | 12,576 | 12,576 | 2.44 (Dev) | 10 |

Table 10: Statistics of multi-hop and single-hop question answering datasets.

**First Iteration:**

1. Generate a summary of source documents to answer the question. Ensure the summary is under 200 words and does not include any pronouns. DO NOT make assumptions or attempt to answer the question; your job is to summarize only.

2. Evaluate the summary based solely on the information of it, without any additional background context: if it lacks sufficient details to answer the question, print [INCOMPLETE]. If it provides all necessary details, print [COMPLETE]. You should provide the reason of the evaluation.

Question: [QUESTION]

Source documents: [SOURCE DOCUMENTS]

Summary:

**Subsequent Iterations:**

1. Generate a summary of the source documents and the previous summary to answer the question based on the evaluation of the previous summary. The evaluation indicates the missing information needed to answer the question. Ensure the summary is under 200 words and does not include any pronouns. DO NOT make assumptions or attempt to answer the question; your job is to summarize only.

2. Evaluate the summary based solely on the information of it, without any additional background context: if it lacks sufficient details to answer the question, print [INCOMPLETE]. If it provides all necessary details, print [COMPLETE]. You should provide the reason of the evaluation.

Question: [QUESTION]

Evaluation of previous summary: [EVALUATION OF PREVIOUS SUMMARY]

Previous summary: [PREVIOUS SUMMARY]

Source documents: [SOURCE DOCUMENTS]

Summary:

Table 11: Prompts used in COMPACT

Source sentences: [SOURCE SENTENCES]

Reference sentences: [REFERENCE SENTENCES]

Question: [QUESTION]

Follow instructions below.
1. Choose 0 to 3 sentences that directly address the critical points needed to answer the question. Additionally, include 0 to 3 sentences that provide useful context, even if they do not directly answer the question. Ensure that you avoid selecting multiple sentences with overlapping content. (prefix: Sentences)

2. Generate a summary of reference sentences and chosen sentences (prefix: Summary). Ensure the summary is under 200 words and does not include any pronouns. DO NOT make assumptions or attempt to answer the question; your job is to summarize only.

3. Evaluate the summary based solely on the information of it, without any additional background context: if it lacks sufficient details to answer the question, print [INCOMPLETE]. If it provides all necessary details, print [COMPLETE]. You should provide the reason of evaluation (prefix: Evaluation)

Sentences:

Table 12: Prompt for dataset construction

**Question**: Who was the eldest brother of the Mexican drug trafficker born 12 March 1952?

**Model**: COMPACT (Ours)

**Iter 1**

**Document (Rank 3)**: Benjamín Arellano Félix (born 12 March 1952) is a Mexican drug trafficker and former leader of the Mexican criminal organization known as the Tijuana Cartel or "Arellano-Félix Organization". Benjamín Arellano Félix, who worked closely with his brothers, was one of Mexico's most powerful drug lords and the supplier of one-third of the U.S.'s cocaine. Benjamín had six brothers: He also has four sisters. Two of them, Alicia and Enedina, are most active in the cartel's affairs. Benjamín was first arrested on 18 June 1982, in Downey, California, for receiving 100 kilos of cocaine smuggled through the San

**Compressed Text**: Benjamín Arellano Félix, born on 12 March 1952, is a Mexican drug trafficker and former leader of the Tijuana Cartel. He had six brothers, including He, and four sisters. He also has four sisters.

**Evaluation**: The summary provides information about Benjamín Arellano Félix and his siblings but does not specify who the eldest brother is. **[INCOMPLETE]**

**Iter 2**

**Document (Rank 7)**: Francisco Rafael Arellano Félix (24 October 1949 - 18 October 2013) was a Mexican drug lord and former leader of the Tijuana Cartel, a drug trafficking organization. He was the oldest of seven brothers and headed the criminal organization early in the 1990s alongside them. Through his brother Benjamín Francisco Rafael joined the Tijuana Cartel in 1989 following the arrest of Miguel Ángel Félix Gallardo, one of the most prominent drug czars in Mexico during the 1980s. When the Arellano Félix took control of the organization in the early 1990s, tensions with the rival Sinaloa Cartel

**Compressed Text**: Francisco Rafael Arellano Félix, born on 24 October 1949, was a Mexican drug lord and former leader of the Tijuana Cartel. He was the oldest of seven brothers and headed the criminal organization early in the 1990s alongside them. Benjamín Arellano Félix, born on 12 March 1952, is a Mexican drug trafficker and former leader of the Tijuana Cartel. He had six brothers, including He, and four sisters. He also has four sisters.

**Evaluation**: The summary provides all necessary details to answer the question. It identifies Francisco Rafael Arellano Félix as the oldest brother of Benjamín Arellano Félix, who was born on 12 March 1952. **[COMPLETE]**

**Answer**: Francisco Rafael Arellano Félix (**Correct**)

**Model**: RECOMP (Xu et al., 2024)

**Summary**: Miguel Rodríguez Orejuela He is the younger brother of Gilberto Rodríguez Orejuela.
Roberto de Jesús Escobar Gaviria Roberto de Jesús Escobar Gaviria Roberto de Jesús Escobar Gaviria (born January 13, 1947), nicknamed El Osito (Little Bear), was the brother of the drug kingpin, Pablo Escobar, and the former accountant of the Medellín Cartel, which was responsible for up to 80 percent of the cocaine smuggled into the United States.

**Answer**: Fabio Ochoa Vásquez (**Wrong**)

Table 13: Example of COMPACT and comparison with RECOMP