# SARA: Selective and Adaptive Retrieval-augmented Generation with Context Compression

**Yiqiao Jin**[1], **Kartik Sharma**[1], **Vineeth Rakesh**[2], **Yingtong Dou**[2], **Menghai Pan**[2], **Mahashweta Das**[2], **Srijan Kumar**[1]
[1]Georgia Institute of Technology, [2]Visa Research
{yjin328,ksartik,srijan}@gatech.edu
{vinmohan,yidou,menpan,mahdas}@visa.com
https://ahren09.github.io/SARA/

## Abstract

Retrieval-augmented generation (RAG) extends large language models (LLMs) with external knowledge, but it must balance limited effective context, redundant retrieved evidence, and the loss of fine-grained facts under aggressive compression. Pure compression-based approaches reduce input size but often discard fine-grained details essential for factual accuracy. We propose SARA, a hybrid RAG framework that targets answer quality under fixed token budgets by combining natural-language snippets with semantic compression vectors. SARA retains a small set of passages in text form to preserve entities and numerical values, compresses the remaining evidence into interpretable vectors for broader coverage, and uses those vectors for iterative evidence reranking. Across 9 datasets and 5 open-source LLMs spanning 3 model families (Mistral, Llama, and Gemma), SARA consistently improves answer relevance (+17.71), answer correctness (+13.72), and semantic similarity (+15.53), demonstrating the importance of integrating textual and compressed representations for robust, context-efficient RAG.

## 1 Introduction

Large language models (LLMs) are powerful, but their knowledge is constrained by the scope, domain, and recency of their training data (Liu et al., 2025). Retrieval-augmented generation (RAG) (Lewis et al., 2020) alleviates this limitation by incorporating external non-parametric knowledge, making it especially useful for knowledge-intensive tasks.

Despite its promise, RAG still faces key challenges in effectively retrieving, selecting, and integrating external evidence. *1) Limited Effective Context.* While some LLMs can process long inputs, their attention is biased toward earlier tokens (Li et al., 2024b), making them sensitive to input order and prone to overlooking important information near the end of the input (Yu et al., 2024; Wang et al., 2026). Expanding the effective context window typically requires costly, model-specific architectural changes (Ding et al., 2023). *2) Context Redundancy.* Retrieved documents often include redundant or loosely structured content, such as meeting transcripts or news articles (Yu et al., 2024; Ge et al., 2024). Without careful post-processing, duplicate or irrelevant content inflates token usage, distracts the model, degrades answer quality or even leads to hallucinations. *3) Compression-Fidelity Trade-off.* Existing context compression techniques can achieve high compression rates, reducing input length, but often at the cost of fine-grained details such as numeric values, organization names, and geographic locations, leading to hallucinated or incomplete responses. As compression becomes more aggressive, critical evidence may be lost, leading to inaccurate responses.

**This Work.** We present SARA, a unified RAG framework that improves both *retrieval* and *generation* stages through structured evidence compression and adaptive selection. SARA employs a two-stage training procedure: 1) *Compression Learning* trains a lightweight compressor in an auto-encoding manner and enables the LLM to reconstruct original text passages from compressed vectors. These vectors–derived from state-of-the-art embedding models (Meng et al., 2024; Muennighoff et al., 2023)–are aligned with the LLM's token embedding space and preserve semantic content while significantly reducing token usage. 2) *Instruction-tuning* adapts the model to reason over mixed-format inputs–natural language and compression vectors–enabling the LLM to balance *local* information (natural language) with global context (compression vectors), achieving a balance between precision and coverage under strict context budgets. At inference, SARA retains the top-$k$ passages in full and compresses the rest into seman-



*Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14508–14528

tically rich, question-agnostic vectors. These serve as lightweight summaries that preserve essential information.

From the *retrieval* perspective, SARA leverages the compression vectors to implement an iterative evidence selection mechanism that dynamically refines the set of top-ranked documents. This mechanism progressively selects contexts based on their incremental value to model understanding, considering both relevance to the query and novelty compared to previously selected evidence. SARA minimizes redundancy while maximizing informativeness through embedding-based novelty and conditional self-information metrics. SARA is agnostic to the choice of embedding models, open-source LLMs, and retrievers. Our contributions are:

- We propose SARA, a novel RAG framework for long-context tasks. SARA introduces a **hybrid compression strategy**, balancing *local precision* using natural language spans and *global abstraction* via compression vectors, enabling fine-grained reasoning and holistic understanding within strict context budgets.
- We propose an **iterative context refinement** mechanism based on the compression vectors to dynamically optimize the retrieved context by reducing redundancy and prioritizing query-relevant content.
- Comprehensive experiments on 5 LLMs spanning 3 model families, including Mistral-7B, MistralNemo-12B, MistralSmall-24B, Llama-3.1-8B, and Gemma3-4B, demonstrate that SARA consistently improves performance (Section 3.2/3.3) and generalizes well across LLMs (Section 3.4) and retrievers (Section 3.5).

## 2 Method

### 2.1 Problem Formulation

A retrieval-augmented generation (RAG) pipeline consists of a *retriever* $\mathcal{R}$ that fetches relevant evidence from a large-scale corpus based on the input query and a *generative model* $\mathcal{M}$ that synthesizes the evidence. Given a query $q$ and corpus $C$, the retriever $\mathcal{R}$ selects the top-$n$ relevant contexts $\mathcal{S} \subseteq C$, which serve as inputs to $\mathcal{M}$ to answer $q$. For effectiveness, a RAG pipeline may incorporate a *reranking* step to reorder the input documents, prioritizing the most relevant ones for answer generation.

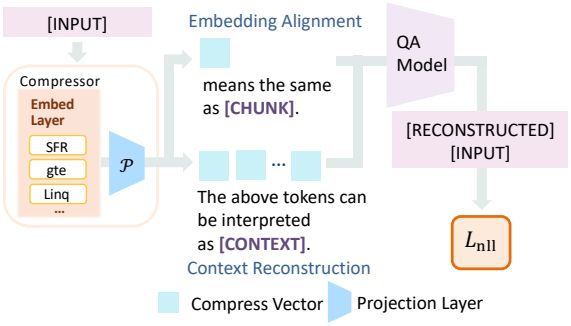


Figure 1: During *Compression Learning*, SARA learns to reconstruct text from compression vectors.

### 2.2 Overview

LLMs have limited effective context windows, and performance degrades when key information is buried in long inputs (Jin et al., 2024b). SARA mitigates this by compressing long context into compact vectors while selectively retaining essential evidence in natural language, preserving model capacity for the most relevant content.

SARA follows a two-stage training procedure: During **Compression Learning**, SARA learns to reconstruct original context from compression vectors. In **Instruction-tuning**, SARA is adapted to rerank the evidence using the compression vectors and reason over mixed inputs–combining natural language and compressed evidence. Our method is *model-agnostic*, compatible with any retrievers, embedding models, and open-source LLMs. A lightweight *projection layer* aligns the embedding space with the LLM space, requiring no significant changes to internal components like the attention mechanism, enabling seamless integration with future embedding models and LLMs. Sample prompts for all stages are in Appendix Table 7.

### 2.3 Compression Learning

An effective compression mechanism should meet three core principles: 1) *Semantic Fidelity*–preserving sufficient information for accurate context reconstruction; 2) *Token Compatibility*–producing compression vectors interpretable by LLMs via prompting; and 3) *Scalability*–requiring minimal adaptation across retrievers and LLMs.

To meet these goals, SARA leverages sentence embeddings (Reimers and Gurevych, 2019) aligned with the LLM's token space to create compact and interpretable representations that support reasoning under tight context budgets.

**Embedding Alignment.** SARA trains a compressor $f$ that encodes each text chunk into a compres-

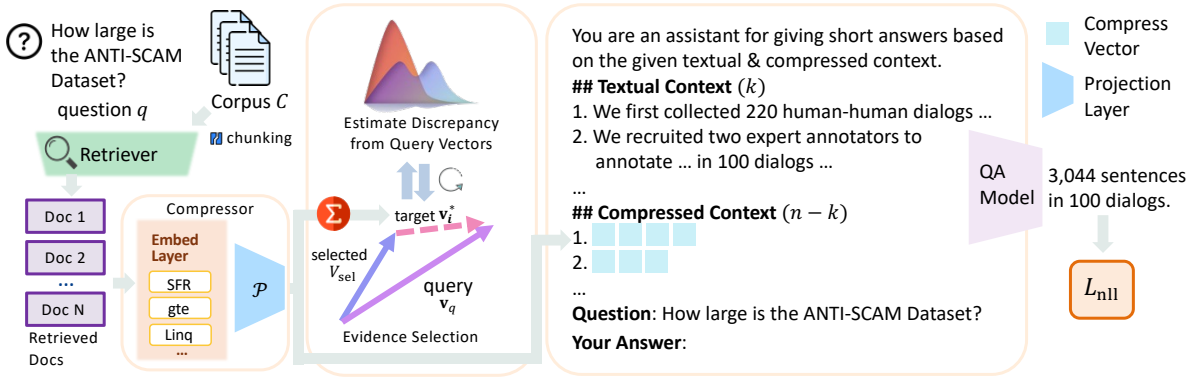


Figure 2: SARA reasons over a mixture of compressed evidence and natural language contexts to balance local precision and global coverage when generating responses. An iterative evidence reranking step selects contexts for relevance and diversity. The retriever, compressor, and QA model uses a variety of embedding models.

sion vector that fits within a single token's embedding space. A lightweight compressor that combines a sentence embedding model and an MLP is trained via an autoencoding task (Liu et al., 2023; Cheng et al., 2024) to align sentence embeddings with the LLM's token space:

$$\mathcal{L}_{\text{a}}(s_i) = -\log P_\theta\left(s_i \mid \phi(s_i), x\right). \quad (1)$$

Here, $\mathcal{L}_a$ is the embedding alignment training objective, $s_i$ is a text chunk, $\phi(\cdot)$ is the compressor, $\theta$ is the model's parameter, and $x$ is the decoding instruction such as "The token <C> can be interpreted as: [CHUNK]." As one compression vector has limited representation capacity, we segment each document into chunks, and encode each chunk as a separate compression vector. We adopt a *curriculum learning* strategy (Bengio et al., 2009; Wang et al., 2021) to improve training stability (Appendix A.1).

**Context Reconstruction.** After learning to decode individual compression vectors, we extend the model to full context reconstruction:

$$\mathcal{L}_r(c) = -\log P_\theta(c \mid \{\phi(s_i), \forall s_i \in c\}, x), \quad (2)$$

where $\mathcal{L}_r$ is the reconstruction loss, $c$ is a document composed of multiple chunks $\{s_i\}$, each encoded as a separate vector. Unlike traditional extractive or abstractive summarization methods (Xu et al., 2024) that require multiple passes, these vectors naturally serve as high-compression-ratio, parallelizable summaries.

**Training Corpus Selection.** To align embedding spaces in a task-agnostic manner, we pretrain on the Wikipedia dataset (Izacard et al., 2023), which provides broad topical diversity and diverse narrative styles, and has proven effective for language model pretraining (Gao et al., 2023). Our compression mechanism preserves information independently of any specific task or query. This enables the LLM to interpret and utilize the compressed representations across a wide range of downstream tasks. Only the compressor $\phi(\cdot)$ and the projection layer are updated during this phase, while the LLM parameters remain frozen. Thus, Wikipedia pretraining does not inject downstream-task knowledge into the generator. In general, the pretraining corpus is domain-agnostic and can be drawn from any natural language dataset.

In Appendix Table 9/10, we demonstrate that these compression vectors are able to encode detailed information, such as exact organization names, academic terms, and numeric values.

### 2.4 Instruction-tuning and Inference

Simple 'retrieve-and-read' pipelines often implies redundant evidence and overlook interdependencies between previously retrieved and newly needed information (Wang et al., 2024). In long-context understanding, *what* should be retrieved next hinges on *what* has already been inferred from previously retrieved evidence (Sarthi et al., 2024; Li et al., 2024a). To address this, SARA leverages a 2-stage context refinement, which interleaves *retrieval* and *reasoning*: 1) a *coarse* retrieval step eliminating irrelevant documents while maintaining computational efficiency; 2) a *fine-grained* reranking step that iteratively refines contexts for informativeness, relevance, and diversity.

**Instruction-tuning.** SARA is instruct-tuned to reason over mixed formats–the top-$k$ passages are input as natural text, while the remaining $n-k$ are encoded as compression vectors (Figure 2). The model generates the answer $a$ conditioned on the

question $q$ and both context types:

$$a = \mathcal{M}(q, \mathcal{S}, \{\phi(s_i), \forall s_i \in \mathcal{V} \backslash \mathcal{S}\})), \quad (3)$$

For training efficiency, we instruct-tune $\mathcal{M}$ on downstream tasks with LoRA (Hu et al., 2021) using top-$n$ contexts retrieved via BM25 (Robertson et al., 2004).

**Dynamic Evidence Reranking.** Effective RAG requires balancing *relevance*–which ensures alignment with the user query–and *novelty*–which introduces new information beyond existing evidence. To achieve this, we adopt an iterative evidence selection method (Algorithm 1) that dynamically selects context based on its incremental value to model understanding.

*Embedding-based Novelty* ranks candidates based on their contribution to the model's discrepancy in knowledge, selecting the vector that minimizes the discrepancy between the selected set $\mathcal{S}$ with query representation $\mathbf{v}_q$ in the embedding space:

$$\mathbf{v}_{\text{agg}} = \text{Agg}\left(\{\phi(v) \mid v \in \mathcal{S} \cup \{v_i\}\}\right), \quad (4)$$

$$\text{Sel}(q, \mathcal{S}, \mathcal{V}) = \underset{v_i \in \mathcal{V} \backslash \mathcal{S}}{\text{argmin}} \left\| \mathbf{v}_q - \mathbf{v}_{\text{agg}} \right\|_2, \quad (5)$$

where $\text{Sel}(\cdot)$ is the evidence selection algorithm. $\text{Agg}(\cdot)$ is the aggregation function, which we implement as average pooling. Since user queries are often brief and may lack sufficient semantic context for effective embedding-based novelty calculation, we supplement the query representation $\mathbf{v}_q$ by aggregating the embeddings of both the question and the top-1 retrieved context: $\mathbf{v}_q = \text{Avg}(\phi(q), \phi(v_1))$. This approach enriches the query representation with domain-specific context from the most relevant retrieved passage, providing a more robust baseline for measuring the novelty of subsequent evidence.

*Conditional Self-information (CSI).* An alternative is to select evidence based on CSI (Shannon, 1948), which quantifies the surprisal of new evidence given previously selected evidence:

$$\text{Sel}(q, \mathcal{S}, \mathcal{V}) = \underset{v_i \in V \backslash \mathcal{S}}{\text{argmax}}\, I(v_i|\mathcal{S}) \quad (6)$$

$$I(v_i|\mathcal{S}) = \frac{1}{|v_i|} \sum_{j=1}^{|v_i|} -\log P(w_i^j \mid \mathcal{S}, w_i^{<j}) \quad (7)$$

where $\text{Sel}(\cdot)$ is the evidence selection function, $I(v_i|\mathcal{S}) = -\log P(v_i|\mathcal{S})$ is the conditional self-information of context $v_j$ given selected contexts $\mathcal{S}$, estimated using a smaller proxy language model. Higher CSI introduces novel information, while lower CSI suggests redundancy with previously selected content. Filtering low-CSI candidates reduces repetition and enhances context *diversity* with minimal impact on overall informativeness.

**Algorithm 1** Query Expansion and Novelty-Based Evidence Selection.

**Input:** Corpus $\mathcal{C} = \{v_i\}_{i=1}^{|\mathcal{C}|}$, query $q$, number of top contexts $n, k$

**Output:** Ranked evidence set $\mathcal{S}$

$\mathcal{V} = \text{Retriever}(q, \mathcal{C})$ ▷ Retrieve top $n$ contexts.

$\mathbf{v}_q = \text{Avg}(\phi(q), \phi(v_1))$ ▷ Enrich query embedding with top-1 context for robust novelty calculation.

$\mathcal{S} \leftarrow \{v_1\}$ ▷ Initialize the set of selected contexts.

**for** $j = 2$ to $k$ **do**

  $\hat{\mathbf{v}} = \text{Agg}(\phi(v), v \in \mathcal{S})$ ▷ Aggregate embeddings of $\mathcal{S}$.

  $v_i^* = \text{Sel}(q, \mathcal{S}, \mathcal{V})$ ▷ Evaluate and select context via Eq. 5 or 6.

  $\mathcal{S} \leftarrow \mathcal{S} \cup \{v_i^*\}$ ▷ Update the selected context set.

**end for**

**return** $\mathcal{S}$

## 3 Evaluation

### 3.1 Experimental Setup

**Models** We evaluate SARA's generalizability across diverse retrieval, embedding, and generation components. For *retrieval*, we experiment with both sparse and dense retrievers, including BM25 (Robertson et al., 2004), `bge-reranker-v2-m3` (Li et al., 2023a) and `SFR-Embedding` (Meng et al., 2024). For the *generation* module, we experiment with 5 LLMs spanning 3 model families: Mistral-7B, MistralNemo-12B, MistralSmall-24B, Llama-3.1-8B, and Gemma3-4B.

**Baselines** We compare our methods with 8 baselines spanning 3 categories: 1) *Standard RAG* (Lewis et al., 2020), which directly feed retrieved documents to the input prompt; 2) *Compression-based methods*, which condense input passages before feeding them into the LLM, including LLMLingua (Jiang et al., 2023b), LongLLMLingua (Jiang et al., 2024), ICAE (Ge et al., 2024), CompAct (Yoon et al., 2024), and xRAG (Cheng et al., 2024); 3) *Summarization-based methods*, which generate intermediate sum-

maries over retrieved documents to support more focused reasoning, including Raptor (Sarthi et al., 2024), GraphRAG (Edge et al., 2024), and InstructRAG (Wei et al., 2025).

**Datasets** We evaluate our approach across diverse datasets spanning different domains, input length, and task types: 1) *Short-context question answering*, including SQuAD-v2.0 (Rajpurkar et al., 2018) 2) *Long-context question answering*, which requires responses based on a single long document, including NarrativeQA (Kočiskỳ et al., 2018), QASPER (Dasigi et al., 2021), QuALITY (Pang et al., 2022), and MultifieldQA-en (Bai et al., 2024); 3) *Multi-hop reasoning*, which requires multi-hop inference across documents, including HotpotQA (Yang et al., 2018), TriviaQA (Joshi et al., 2017), 2WikiMultihopQA (Ho et al., 2020); 4) *Summarization*, including QMSum (Zhong et al., 2021). We use SQuAD-v2, NarrativeQA, QASPER, QuALITY, HotpotQA, and TriviaQA for both training and evaluation. MultifieldQA-en, 2WikiMultihopQA, and QMSum are held out for out-of-domain evaluation only. All corpora are split into 256-token chunks aware of the sentence structures. Detailed dataset descriptions and statistics are in Appendix A.2.

**Metrics** We adopt standard evaluation protocols consistent with prior work (Asai et al., 2023; Cheng et al., 2024; Sarthi et al., 2024; Edge et al., 2024). For holistic evaluation, we report both traditional *lexical metrics*–including ROUGE (R-L) (Lin, 2004), F1 match scores–and *LLM-based metrics* (Es et al., 2024), including response relevance, answer correctness, semantic similarity, and faithfulness. Full metric definitions and implementation details are in Appendix A.3.

**Settings** Compression- (Type 2) and Summarization methods (Type 3) differ in both mechanism and computational assumptions. Summarization methods rely on large LLMs for thematic abstraction, while compression methods retain factual content in compact representations suitable for models with varying sizes. To ensures fairness and clarity, we evaluate them separately to gauge both accuracy and efficiency–summarization methods are tested without input limits, whereas compression methods are assessed under strict token budgets:

**S1. General QA Performance** assesses overall answer accuracy when the models are not constrained by input length. It benchmarks SARA against summarization-based methods such as Raptor and GraphRAG, both of which leverage the more powerful GPT-4o (OpenAI, 2025) as the model for question-answering $\mathcal{M}$ and hierarchical summarization, as open-source models struggle with reasoning over long complex inputs. This comparison highlights SARA 's capability to achieve strong open-domain QA accuracy using smaller models and without relying on costly global summarization. Unless otherwise mentioned, we retrieve $n = 10$ contexts per query, retain the top $k = 5$ in natural language, and compress the remaining. **S2. Context-Efficiency Evaluation** isolates SARA 's efficiency under strict input budgets. Here, we compare against leading compression-based methods using a fixed QA backbone (Mistral-7B). This setup directly measures how effectively each method preserves essential information when context length is limited–a critical requirement for scalable or edge-device deployment. The input context length is constrained to 512 and 1024 tokens to evaluate performance under strict context budgets.

### 3.2 S1. General QA Performance

Compared with summarization-based methods (Table 1), SARA consistently outperforms standard RAG and state-of-the-art summarization-based baselines, including Raptor and GraphRAG, despite their use of the much stronger GPT-4o (OpenAI, 2025) as backbones for question-answering and summarization. On HotpotQA, which requires multi-hop reasoning, SARA achieves +15% F1 and +14.6% ROUGE-L. These results highlight the effectiveness of our compression approach in helping the model accommodate and reason over multiple discrete contexts within constrained context.

### 3.3 S2. Context-Efficiency Evaluation

Figure 3 and Appendix Figure 7 compare SARA and strong compression-based methods under context length constraints (512 and 1024 tokens). SARA consistently outperforms baselines on both lexical (F1, ROUGE-L) and LLM-based evaluation metrics.

Under 512 tokens, SARA improves F1 by 19.4% and ROUGE-L by 20.8% on average. We observe that the gains are particularly significant on knowledge-intensive tasks like TriviaQA (+24.5%) and HotpotQA (+29.0%), which require facts and reasoning. Improvements on narrative-style tasks (e.g. NarrativeQA) are more modest, particularly under 1024 tokens (+6.6% F1 and 6.8% ROUGE-L), likely because chunking and compres-

| Dataset | QASPER | | NarrativeQA | | TriviaQA | | QuALITY | | HotpotQA | |
|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | F1 | R-L | F1 | R-L | F1 | R-L | F1 | R-L | F1 | R-L |
| RAG | 22.73 | 16.71 | 40.23 | 40.16 | 58.43 | 49.07 | 31.79 | 31.63 | 48.56 | 40.06 |
| Raptor | 31.77 | 25.26 | 56.60 | 56.91 | 70.51 | 65.46 | 34.27 | 34.49 | 68.26 | 63.14 |
| GraphRAG | 37.05 | 36.66 | 64.93 | 63.55 | 77.52 | 72.35 | 37.21 | 38.15 | 73.23 | 68.21 |
| xRAG | 32.36 | 33.72 | 33.43 | 32.15 | 43.36 | 35.52 | 32.65 | 33.84 | 60.19 | 49.56 |
| InstructRAG | 32.83 | 33.92 | 41.79 | 39.85 | 76.47 | 72.19 | 37.98 | 38.30 | 66.77 | 60.18 |
| SARA-CSI | 38.83 | 41.52 | **69.46** | **68.02** | **85.08** | 83.85 | **42.78** | 44.18 | **84.21** | **78.16** |
| SARA-EMB | **40.55** | **41.71** | 69.15 | 66.55 | 84.74 | **84.17** | 42.59 | **44.31** | 83.77 | 76.37 |
| *Impr. %* | 9.4% | 13.8% | 7.0% | 7.0% | 9.8% | 16.3% | 12.6% | 15.7% | 15.0% | 14.6% |

Table 1: General QA performance (S1) of SARA, vanilla RAG, and state-of-the-art summarization-based methods when using Mistral-7B (Jiang et al., 2023a) as the QA model $\mathcal{M}$ and BM25 (Robertson et al., 2004) as retriever $\mathcal{R}$.

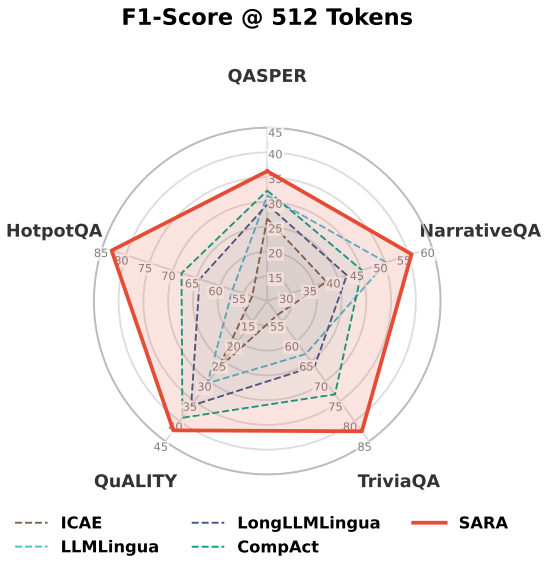


Figure 3: SARA and compression-based methods' performance under token budgets of 512. All methods use Mistral-7B (Jiang et al., 2023a) as the QA model $\mathcal{M}$.

sion can change the narrative flow and obscure subtle discourse-level cues. Unlike factoid questions, narrative questions demand holistic coherence that is harder to retain under chunking and summarization (Ge et al., 2024).

**Impact of Context Budgets** Increasing the context budget from 512 to 1024 tokens generally improves performance. Baselines that produce natural language compression (e.g., LongLLMLingua) see substantial gains–up to +10.6 F1 on NarrativeQA–as the additional budget reduces the need to truncate or overly compress passages, allowing inputs to better reflect their original structure. SARA retains a clear performance lead, outperforming the strongest baseline by 6-12 F1 on knowledge-intensive tasks such as TriviaQA and HotpotQA. As SARA has already captured key content efficiently under a lower context budget through hybrid compression, it exhibits relatively modest gains on certain datasets (e.g., +4.1 F1 on QASPER).

**Balancing Compression Efficiency and Answer Faithfulness** A central challenge in RAG is balancing compression efficiency with faithfulness. Aggressive approaches like xRAG, which compress entire evidence sets into a single dense vector, optimize for efficiency but often at the cost of factuality and hallucination. As shown in Table 1, baselines like xRAG may struggle on knowledge-intensive tasks. Qualitative analysis in Appendix Table 8 reveals that baselines can hallucinate content, generating answers with fabricated entities or tasks ('sentiment analysis' and 'machine translation') ungrounded in the original documents. Methods that over-compress inputs (e.g. ICAE) risk discarding critical content. As a result, the model tends to become overly conservative-frequently concluding that the answer is not present. These failures underscore the drawbacks of one-shot compression when multiple facts must be retained. In contrast, SARA can accurately recovers fine-grained content, such as specific task names (e.g. NLI, document and intent classification) prompted in the question) with high fidelity, even under tight context budgets. Thus, SARA's hybrid approach preserves salient content, simplifying key information while mitigating factual distortion under tight context budgets.

**Performance on Short-context QA** SQuAD-v2 presents minimal challenges in context length, as each query is paired with a single passage that fits within the model's input window in most cases. Accordingly, the performance gap across models narrows. SARA achieves the highest results (76.55 F1, 69.22 ROUGE-L; Table 3), outperforming the

strongest baseline by a modest margin (+3.98 F1, +2.19 ROUGE-L). In contrast, aggressively compressed systems such as xRAG and ICAE perform significantly worse ($\leq$ 60.19 F1), likely due to summaries that obscures critical details, e.g. entity names, numeric values, and events, reducing accuracy even when full text fits into the model.

### 3.4 Generalizability across LLMs & Sizes

Beyond Mistral-7B, we evaluate SARA on 4 additional models from 3 families–Mistral, Llama, and Gemma–spanning various sizes and architectures: MistralNemo-12B, MistralSmall-24B, Llama3.1-8B, and Gemma3-4B. As shown in Figures 4& 8, SARA consistently outperforms the baseline, with up to +40 in Answer Relevance, +14 in Answer Correctness, and +21 in Semantic Similarity. Improvements are particularly pronounced on smaller models. On Mistral-7B, SARA boosts answer relevance by 17.71, answer correctness by 13.72, and semantic similarity by 15.53. These results highlight the method's ability to optimize context usage under tighter context budgets, making it especially effective for smaller models. In some cases, SARA enables a 7B model to match or surpass much larger ones (e.g., MistralSmall-24B), highlighting that reasoning over mixed-format contexts can close the performance gap without increasing model sizes.

In general, performance gains are more significant when the compressor and LLM share the same architecture (e.g. Mistral). Among the Mistral family, we observe an average boost in Answer Relevance of 20.12 and Answer Correctness of 7.07. MistralNemo and MistralSmall achieve improvements in response relevance of +19.65 and +23.01, and semantic similarity of +20.44 and +14.38, respectively. This suggests that architectural alignment between the compressors and LLMs enhances semantic compatibility between compressed inputs and answer generation. In contrast, Gemma-3 shows modest gains (e.g. +6.83 in answer relevance and +5.82 in answer correctness), likely due to its architectural mismatch. Note that SARA does not aim to directly enhance the QA model's intrinsic generation capability. Instead, its strength lies in refining and reorganizing retrieved contexts to support finer-grained reasoning. Since both SARA and RAG leverage the same initial retriever, they operate over comparable evidence. As a result, faithfulness–the factual consistency with the retrieved context–shows modest improvements.

### 3.5 Generalization Across Retrievers

We evaluate SARA with dense retrievers like `multi-qa-mpnet-base-cos-v1` (Song et al., 2020) and SFR (Meng et al., 2024) in addition to BM25 (Robertson et al., 2004). As shown in Table 4, SARA performs consistently across retrievers, confirming its model-agnostic design. Dense retrievers, especially SFR, yield stronger results–achieving +19 F1 over BM25 on QASPER–highlighting the value of semantically richer base retrievers for complex, multi-hop QA. Overall, SARA remains robust to retriever choice while benefiting from higher-quality evidence.

### 3.6 Sensitivity Analysis

We analyze how SARA balances natural language evidence and compression vectors under varying evidence budgets. In Figure 5 and Appendix Figure 10, we fix the total number of retrieved passages ($n = 10$) and vary $k$, the number of top-ranked passages in natural language. As a comparison, we also show the results without compression vectors in the same plots.

SARA consistently outperforms the baseline without compression vectors, particularly when $k$ is small. For QASPER (Figure 5), the gap is largest at $k = 1$ (F1: 36.59 vs. 25.12 on QASPER), underscoring the value of compression in retaining critical information with minimal input length. Performance improves in both settings as $k$ increases, with diminishing gains beyond $k = 8$. At $k = 10$, both approaches converge, suggesting that compression is most beneficial under limited context budgets. This highlights the strength of our hybrid strategy: compression complements full-text evidence, maintaining effectiveness across evidence-rich and resource-constrained scenarios.

**Effects of Varying $k$ & Error Analysis** Table 5 shows how varying $k$ affects factual completeness. With all contexts compressed ($k = 0$), the model captures only coarse information (e.g., identifying CoNLL-2003 but omitting OntoNotes-5.0, MSRA, and Weibo). As $k$ increases, factual recall improves: at $k = 2$, the model adds OntoNotes-5.0 and partial mentions of Chinese datasets, and at $k$=5, it recovers nearly all ground-truth datasets (CoNLL-2003, OntoNotes-5.0, MSRA, Weibo, Resume). These results highlight that SARA's hybrid design–combining compressed vectors with natural language contexts–enables high factual fidelity while maintaining efficiency.

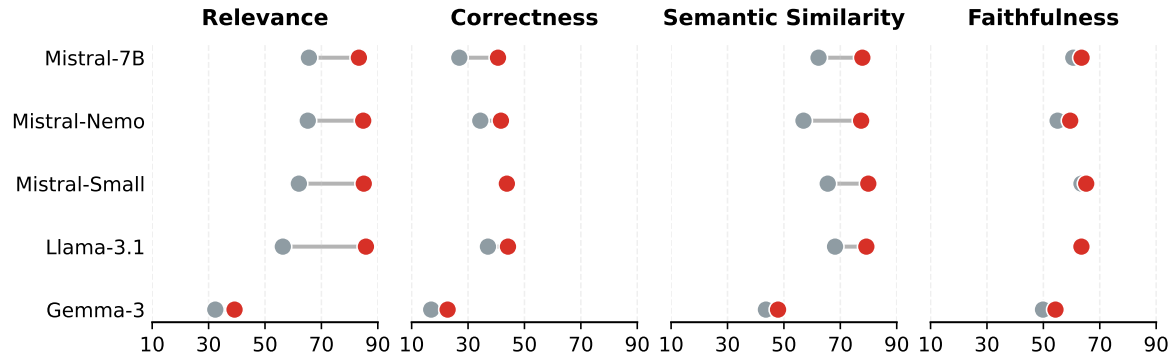


Figure 4: Performance of RAG and SARA across different LLMs in terms of LLM-based metrics on QASPER (Dasigi et al., 2021).

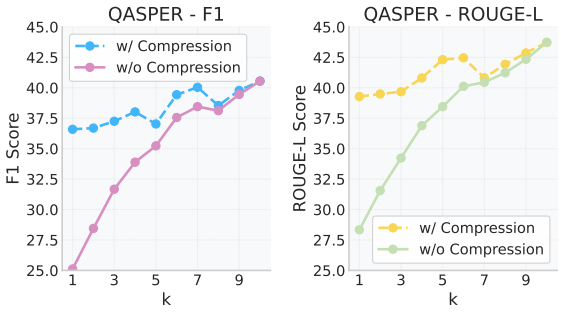


Figure 5: Sensitivity analysis with total contexts fixed at $N = 10$, varying the number of natural language contexts $k$. Performance generally improves as $k$ increases, peaking around $k = 7$-8. The performance gap before and after applying SARA is the largest when $k = 1$, indicating that the compression vectors add to performance gain with minimal addition to input lengths.

**Effects of Varying $n$** increasing $n$ beyond 10 offers diminishing gains. On QASPER, expanding $n$ from 10 to 50 improves F1 from 40.55 to 42.26 and ROUGE-L from 41.71 to 43.09 (with $k$=5), but incurs a substantial rise in computation time. This pattern reflects diminishing returns once the retrieved set sufficiently captures relevant evidence.

**Effects of Shorter Chunk Sizes** Shorter sequence lengths hinder long-form reasoning. On NarrativeQA, reducing the length to 64 tokens lowers F1 from 69.46 to 61.74 due to aggressive chunking that breaks semantic coherence in narratives.

### 3.7 Ablation Studies

To quantify the contribution of each major component–compression, reconstruction, and reranking–we evaluate 3 variants of SARA. **SARA-C** removes the **C**ompression vectors and only process contexts in natural language formats. **SARA-P** removes the context reconstruction objective during training (Section 2.3). **SARA-R** skips the adaptive reranking stage, relying solely on initial BM25 retrieval (Section 2.4). **SARA-L** removes the curriculum learning in embedding alignment (Section 2.3 Equation 1). **SARA-I** removes the instruction-tuning step (Section 2.4).

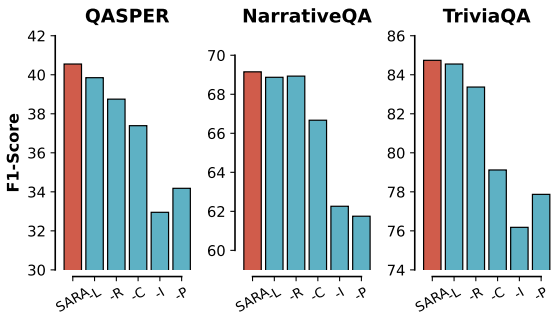


Figure 6: Performance of SARA's variants.

**Context Reconstruction is Critical.** Removing the reconstruction objective (SARA-P) results in the most substantial performance drop (Figure 6)–7-9 F1 across all datasets. This confirms that learning to reconstruct full contexts from compressed vectors is essential for preserving semantic and leveraging these vectors for accurate answer generation. On the contrary, liminating curriculum learning in the embedding alignment module (SARA-L) causes only a small average performance decrease (≈0.8 F1). This indicates that while curriculum learning are effective, the core retrieval and compression mechanisms remain robust.

**Compression Improves Performance.** Disabling compression (SARA-C) leads to consistent performance declines, especially on TriviaQA (-5.6 F1) where the long-form contexts may contain noisy or irrelevant information. These results suggest that compression helps filter salient content and suppress redundancy, leading to improved answer correctness.

**Reranking offers Measurable Gains.** Removing reranking (SARA-R) yields modest but consistent drops, confirming that compression-aware

reranking improves evidence selection beyond lexical similarity–especially when initial retrieval are suboptimal–at minimal computational cost.

**Instruction-Tuning is Essential.** Removing the instruction-tuning step (SARA-I) leads to a substantial degradation (≈7 F1 on average). Although embedding alignment already enhances semantic representation, instruction-tuning is crucial for enabling the generator to effectively interpret and utilize the compression vectors in downstream tasks.

## 4 Related Work

### 4.1 Retrieval-augmented Generation (RAG)

Retrieval-augmented Generation has become a standard practice for knowledge-intensive tasks. Instead of treating LLMs as knowledge repositories, RAG retrieves from an external knowledge base (Lewis et al., 2020; Sharma et al., 2024), mitigating issues about knowledge cutoffs and insufficient training coverage. However, LLMs struggle to process long contexts effectively, even with extended context windows (Wang et al., 2025), highlighting the importance of reranking and evidence structuring. A prominent approach is retrieve-rerank-generate (Yu et al., 2024), where reranking boosts downstream QA performance by prioritizing relevant or diverse contexts (Thakur et al., 2001; Izacard et al., 2023). Methods like Raptor (Sarthi et al., 2024), GraphRAG (Edge et al., 2024) and GraphReader (Li et al., 2024a) improve the *retrieval* and *reranking* stages by structuring retrieved content, enhancing RAG through semantic or graph-based organization of knowledge, leading to more relevant and compact inputs for generation.

### 4.2 Context Compression

Context compression is essential for reducing inference costs and improving efficiency in long-context (Pan et al., 2024; Rau et al., 2025) or multi-turn scenarios (Kim et al., 2024a). Prior work fall into two main directions: natural-language (NL)-based and representation-level compression. *NL-based compression* (Zhang et al., 2024b; Chirkova et al., 2025) like AdaComp (Zhang et al., 2024b), CompAct (Yoon et al., 2024), and EXIT (Hwang et al., 2024) condense prompts or histories into extractive or abstractive summaries. These methods are generally model-agnostic and applicable across open-source and proprietary LLMs (Zhu et al., 2025). *Representation-based methods* (Chevalier et al., 2023; Munkhdalai et al., 2024; Louis et al., 2025b,a) treat LLMs as white boxes and modify attention calculation (Munkhdalai et al., 2024), positional encodings (Jin et al., 2024a; Zhang et al., 2024c), or embeddings (Cheng et al., 2024). Methods such as xRAG (Cheng et al., 2024), GIST (Mu et al., 2023), and ICAE (Ge et al., 2024) embed and compress contexts in the model's latent space.

## 5 Conclusion

We present SARA, a unified and efficient RAG framework that jointly enhances retrieval, reranking, and generation through structured evidence compression and adaptive document selection. Experiments across LLM backbones, retrievers, and embedding models demonstrate that SARA significantly improves answer correctness and relevance.

## 6 Limitations

While SARA demonstrates strong performance, several aspects present opportunities for further improvement and exploration.

First, the fidelity of compressed representations may vary depending on context. In particular, fine-grained details such as numerical values or entity names can be more difficult to preserve after compression. Future work could explore lightweight probing models or evidence reconstruction techniques (e.g., conditional self-information or embedding similarity) to better assess and enhance semantic preservation. Second, the current compression-by-sentence strategy is a simple yet general choice. Adaptive mechanisms that adjust vector granularity based on context complexity could further improve efficiency. Determining the appropriate number of compression vectors per context remains an open challenge. Third, our design requires access to the model's input embeddings. Thus, SARA is most readily applicable to open-source models. Exploring API-compatible variants could be an important future direction to broadening the applicability to proprietary models. Fourth, on narrative-heavy datasets (e.g., NarrativeQA), chunk-level compression can fragment event sequences or entity relations, which limits gains relative to factoid or multi-hop QA. Narrative-aware segmentation heuristics and coreference-resolution-guided boundaries are promising directions for preserving discourse-level coherence under compression.

## References


Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *ICLR*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *ACL*, pages 3119–3137.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*, pages 41–48.

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:2405.13792*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *EMNLP*, pages 3829–3846.

Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. 2025. Provence: efficient and robust context pruning for retrieval-augmented generation. *arXiv:2501.16214*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *NAACL*, pages 4599–4610.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv:2307.02486*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv:2404.16130*.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. Ragas: Automated evaluation of retrieval augmented generation. In *EACL*, pages 150–158.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *EMNLP*, pages 6465–6488. ACL.

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 *ICLR*.

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 *COLING*, pages 6609–6625.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2021. Lora: Low-rank adaptation of large language models. In *ICLR*.

Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C Park. 2024. Exit: Context-aware extractive compression for enhancing retrieval-augmented generation. *arXiv:2412.12559*.

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. *JMLR*, 24(251):1–43.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 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 *EMNLP*, pages 13358–13376.

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 *ACL*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024a. Llm maybe longlm: Selfextend llm context window without tuning. In *ICML*, pages 22099–22114.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024b. Agentreview: Exploring peer review dynamics with llm agents. In *EMNLP*, pages 1208–1226.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, pages 1601–1611.

Jang-Hyun Kim, Junyoung Yeom, Sangdoo Yun, and Hyun Oh Song. 2024a. Compressed context memory for online language model interaction. In *ICLR*.

Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, Jy-yong Sohn, and Chanyeol Choi. 2024b. Linq-embed-mistral:elevating text retrieval with improved gpt data through task-specific control and quality refinement. Linq AI Research Blog.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 33:9459–9474.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023a. Making large language models a better foundation for dense retrieval. *arXiv:2312.15503*.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, and 1 others. 2024a. Graphreader: Building graph-based agent to enhance long-context abilities of large language models. In *EMNLP*, pages 12758–12786.

Wenyan Li, Jiaang Li, Rita Ramos, Raphael Tang, and Desmond Elliott. 2024b. Understanding retrieval robustness for retrieval-augmented image captioning. In *ACL*, pages 9285–9299.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning. *arXiv:2308.03281*.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *NeurIPS*, 36:34892–34916.

Jerry Liu. 2022. LlamaIndex. https://github.com/jerryjliu/llama_index. DOI: 10.5281/zenodo.1234.

Shudong Liu, Yiqiao Jin, Cheng Li, Derek F Wong, Qingsong Wen, Lichao Sun, Haipeng Chen, Xing Xie, and Jindong Wang. 2025. Culturevlm: Characterizing and improving cultural understanding of vision-language models for over 100 countries. *arXiv:2501.01282*.

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025a. Pisco: Pretty simple compression for retrieval-augmented generation. *arXiv:2501.16075*.

Maxime Louis, Thibault Formal, Hervé Dejean, and Stéphane Clinchant. 2025b. Oscar: Online soft compression and reranking. *arXiv:2504.07109*.

Rui Meng, Ye Liu, Shafiq Rayhan Joty, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Jesse Mu, Xiang Li, and Noah Goodman. 2023. Learning to compress prompts with gist tokens. *NeurIPS*, 36:19327–19352.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *ACL*, pages 2014–2037.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv:2404.07143*.

OpenAI. 2025. Gpt-4o.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, and 1 others. 2024. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *ACL*, pages 963–981.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and 1 others. 2022. Quality: Question answering with long input texts, yes! In *NAACL*, pages 5336–5358.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*, pages 784–789.

David Rau, Shuai Wang, Hervé Déjean, Stéphane Clinchant, and Jaap Kamps. 2025. Context embeddings for efficient answer generation in retrieval-augmented generation. In *WSDM*, pages 493–502.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*. Association for Computational Linguistics.

Julian Risch, Timo Möller, Julian Gutsch, and Malte Pietsch. 2021. Semantic answer similarity for evaluating question answering models. In *Proceedings of the 3rd Workshop on Machine Reading for Question Answering*, pages 149–157.

Stephen Robertson, Hugo Zaragoza, and Michael Taylor. 2004. Simple bm25 extension to multiple weighted fields. In *CIKM*, pages 42–49.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *ICLR*.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Kartik Sharma, Peeyush Kumar, and Yunqing Li. 2024. Og-rag: Ontology-grounded retrieval-augmented generation for large language models. *arXiv:2412.15235*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *NeurIPS*, 33:16857–16867.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2001. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks Track*.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, and 1 others. 2024. Searching for best practices in retrieval-augmented generation. In *EMNLP*.

Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576.

Yiyang Wang, Chen Chen, Tica Lin, Vishnu Raj, Josh Kimball, Alex Cabral, and Josiah Hester. 2025. Companioncast: A multi-agent conversational ai framework with spatial audio for social co-viewing experiences. *arXiv:2512.10918*.

Yiyang Wang, Yiqiao Jin, Alex Cabral, and Josiah Hester. 2026. Mascot: Towards multi-agent socio-collaborative companion systems. *arXiv:2601.14230*.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2025. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. In *ICLR*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*, pages 38–45.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. Recomp: Improving retrieval-augmented lms with compression and selective augmentation. In *ICLR*.

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 *EMNLP*, pages 2369–2380.

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

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *NeurIPS*.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2024a. Jasper and stella: distillation of sota embedding models. *arXiv:2412.19048*.

Qianchi Zhang, Hainan Zhang, Liang Pang, Hongwei Zheng, and Zhiming Zheng. 2024b. Adacomp: Extractive context compression with adaptive predictor for retrieval-augmented large language models. *arXiv:2409.01579*.

Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. 2024c. Found in the middle: How language models use long contexts better via plug-and-play positional encoding. In *NeurIPS*.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and 1 others. 2021. Qmsum: A new benchmark for query-based multi-domain meeting summarization. In *NAACL*, pages 5905–5921.

Wenhao Zhu, Pinzhen Chen, Hanxu Hu, Shujian Huang, Fei Yuan, Jiajun Chen, and Alexandra Birch. 2025. Generalizing from short to long: Effective data synthesis for long-context instruction tuning. *arXiv:2502.15592*.

# A Experimental Details

## A.1 Implementation Details

Our implementation is based on `PyTorch` (Paszke et al., 2019), `transformers` (Wolf et al., 2020), and `llama-index` (Liu, 2022). All models and data use the `bfloat16` data type. For LoRA setup, we adopt a rank attention dimension of 16, scaling factor $\alpha = 32$, and dropout of 0.1. For chunking, we set the chunk size to 256. The model processes at most $n = 10$ chunks. Our method further selects the top $k = 5$ as natural language evidence, and encode the rest as compression vectors. To reduce the effects of stochasticity, we fix the sampling temperature at 0. Experiments were performed on a Linux server with 6 NVIDIA A100 GPUs.

For embedding alignment (Section 2.3), we adopt a *curriculum learning* strategy, starting with shorter sentences and gradually transition into complex examples. Specifically, we use spaCy[1] for NER and rank sentences by token count and the number of named entities in categories such as `PER`, `ORG`, `LOC`, `GPE`, `Date`, `Time`, and `Event`.

## A.2 Dataset Descriptions

- NarrativeQA (Kočiskỳ et al., 2018): question-answering based on books and movie transcripts.
- QASPER (Dasigi et al., 2021): information seeking over scientific research papers with supporting evidence spans.

[1] `https://spacy.io/`

| Model | Full Name | Base LLM | Size |
|---|---|---|---|
| SFR (Meng et al., 2024) | `Salesforce/SFR-Embedding-Mistral` | Mistral-7B | 4096 |
| Linq (Kim et al., 2024b) | `Linq-AI-Research/Linq-Embed-Mistral` | Mistral-7B | 4096 |
| GTE (Li et al., 2023b) | `Alibaba-NLP/gte-Qwen2-7B-instruct` | Qwen2-7B | 3584 |
| Stella (Zhang et al., 2024a) | `NovaSearch/stella_en_1.5B_v5` | Qwen2-1.5B | 8960 |

Table 2: Embedding models used in the compressor and their embedding sizes.

- QuALITY (Pang et al., 2022): reading-comprehension benchmark with $\sim$ 5000-token passages and unambiguous questions that require consolidating information from multiple text segments.

- TriviaQA (Joshi et al., 2017): trivia questions paired with web evidence (news, encyclopedia, and blogs).

- HotpotQA (Yang et al., 2018): natural questions that require multi-hop reasoning. The questions are annotated with supporting facts.

- SQuAD-v2.0 (Rajpurkar et al., 2018): questions are based on Wikipedia articles, and the answers are text segments from the corresponding reading passage. We select questions that are marked as "answerable"

- QMSum (Zhong et al., 2021): query-focused meeting summarization from dialogue transcripts.

- MultifieldQA-en (Bai et al., 2024) single-doc QA from diverse sources (arXiv, C4, Wikipedia, WuDaoCorpora, etc.)

- 2WikiMultihopQA (Ho et al., 2020): multi-hop QA combining structured and unstructured evidence with reasoning paths.

### A.3 Evaluation Metrics

**Automatic Evaluation.** For free-form answer generation, we report ROUGE-L (R-L) (Lin, 2004) and F1 match scores to measure lexical overlap between predicted and ground-truth answers.

**LLM-based Evaluation.** To complement traditional lexical scores, we adopt four LLM-based metrics that capture orthogonal dimensions essential for reliable RAG deployment (Es et al., 2024; Risch et al., 2021). Each metric returns a value in $[0, 1]$, with higher values indicating better performance.

- **Faithfulness** measures whether the generated answer is grounded in the retrieved context. The answer is decomposed into atomic claims with GPT-4o. Each claim is then tested for entailment against the retrieved context. Answers fully supported by the evidence are favored, and hallucinations are penalized.

- **Answer Relevance** (Response Relevance) judges how directly the answer addresses the user's question. Redundant, off-topic, or missing information lowers the score. It does not take factual accuracy into consideration.

- **Factual Correctness** uses claim decomposition and natural language inference to verify the model's claims against reference texts.

- **Semantic Similarity** uses a cross-encoder to compute the semantic overlap between the generated answer and the ground-truth reference.

## B Additional Experiments

| Dataset | SQuAD-v2 | |
|---|---|---|
| Metrics | F-1 | R-L |
| RAG | 63.65 | 51.26 |
| Raptor | 70.69 | 65.28 |
| GraphRAG | 74.82 | 67.36 |
| xRAG | 60.19 | 49.56 |
| InstructRAG | 67.21 | 57.94 |
| ICAE | 50.31 | 40.82 |
| LLMLingua | 70.24 | 65.12 |
| LongLLMLingua | 72.57 | 67.03 |
| SARA | 76.55 | 69.22 |

Table 3: Performance comparison on the SQuAD-v2 dataset.

### B.1 Generalization on Unseen Datasets

We evaluate out-of-domain generalization on three LongBench datasets (Bai et al., 2024):

| Retriever | QASPER | | NarrativeQA | | TriviaQA | |
|---|---|---|---|---|---|---|
| | F-1 | ROUGE-L | F-1 | ROUGE-L | F-1 | ROUGE-L |
| SFR | 55.44 | 52.93 | 58.03 | 56.39 | 84.13 | 83.61 |
| BGE | 44.47 | 45.24 | 54.05 | 53.98 | 85.41 | 84.58 |
| BM25 | 36.15 | 39.54 | 56.79 | 55.76 | 83.58 | 83.65 |

Table 4: Generalizability across different retrievers.

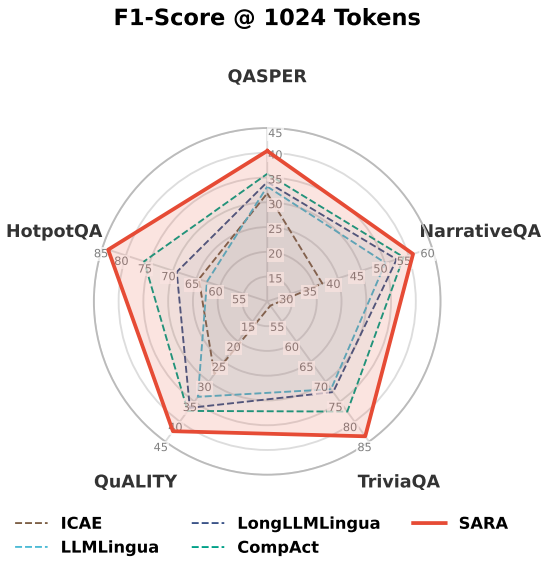


Figure 7: Performance of SARA and compression-based methods under token budgets of 1024. All methods use Mistral-7B (Jiang et al., 2023a) as the QA model $\mathcal{M}$.

MultiFieldQA-en, 2WikiMultihopQA, and QMSum, which differ substantially from the training data in both domain and task format.

As shown in Table 6, SARA consistently improves performance across all three benchmarks, with especially large gains in RESPONSE RELEVANCE: +18.6 on QMSum, +47.7 on MultiFieldQA-en, and +55.0 on 2WikiMultihopQA. This suggests that combining natural language spans with compression vectors helps the model identify and use more relevant evidence even under domain shift, leading to more focused and less off-topic responses.

In contrast, gains in ANSWER CORRECTNESS are smaller (+0.3 to +2.2). This gap suggests that the retrieval and grounding benefits of SARA transfer more readily than the downstream reasoning required to produce fully correct answers. Overall, these results indicate that SARA generalizes robustly as an evidence selection mechanism, while leaving additional room for improvement in domain-specific reasoning and generation.

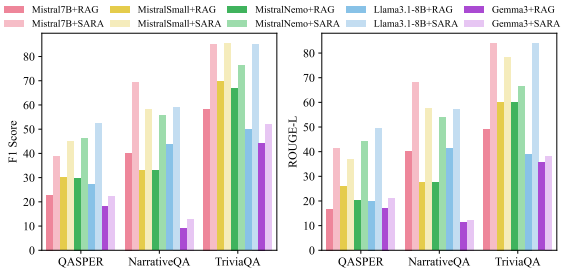


Figure 8: Generalizability across models. We report lexical metrics (F1 & ROUGE-L) on QASPER (Dasigi et al., 2021) before / after applying SARA.

## B.2 Efficiency & Latency Analysis

For training costs, our compression learning phase requires approximately 36 hours on 4 NVIDIA A100 GPUs for Wikipedia pretraining, which is a one-time cost that enables cross-task generalization.

For inference costs, we evaluate the runtime and resource efficiency of SARA compared to standard RAG (using either SFR or BM25 as retriever) and baseline methods. The results are summarized in Table 11.

**Minimal Overhead under Standard Settings** When reranking a single context ($k = 1$), SARA introduces minimal latency overhead relative to standard RAG. Its end-to-end latency is 1.72 seconds, comparable to 1.70 seconds and 1.68 seconds for RAG with SFR and BM25, respectively. Host memory usage remains moderate (2546,MB vs. 2233,MB for RAG-SFR), confirming that the reranking pipeline imposes minimal additional cost.

**Stable Resource Footprint with Increased Context.** Increasing the number of reranked passages to $k = 5$ slightly raises latency (to 1.98 seconds) and host memory usage (from 2546 MB to 2741 MB). When changing the retriever from BM25 to SFR, the GPU memory usage increases from 15.5 GB for RAG (BM25) to 26.4 GB RAG (SFR), as the SFR embedding requires GPU memory. Because SARA reuses the same SFR encoder as both

| Question | Which NER dataset do they use? |
|---|---|
| Evidence | • CoNLL2003 is one of the most evaluated English NER datasets, which contains four different named entities: PERSON, LOCATION, ORGANIZATION, and MISC ...<br>• OntoNotes 5.0 is an English NER dataset whose corpus comes from different domains, such as telephone conversation, newswire. We exclude ...<br>• ... OntoNotes 4.0 ... we use the Chinese part. We adopted the same pre-process ...<br>• The corpus of the Chinese NER dataset MSRA came from news domain ...<br>• Weibo NER was built based on text in Chinese social media Sina Weibo, and it contained 4 kinds of entities ...<br>• Resume NER was annotated by ... |
| Ground-truth | The datasets include CoNLL2003, OntoNotes 5.0, OntoNotes 4.0, the Chinese NER dataset MSRA, Weibo NER, and Resume NER. |
| Predictions | |
| 0/10 | They use the **CoNLL-2003 NER dataset**. |
| 2/8 | The NER dataset they use is **CoNLL-2003, OntoNotes-5.0** and data based on Chinese social media. |
| 5/5 | The NER datasets used are **CoNLL-2003, OntoNotes-5.0, MSRA, Weibo, and Resume.** |

Table 5: Sample responses when using `Llama-3.1-8B-Instruct` as the base model with varying numbers of natural language and compressed contexts. '2/8' means using 2 natural language and 8 compressed context. Exact matches with the ground-truth answer is in **bold** and semantic similar parts are in gray. As the number of natural language contexts increase, the model answers are more detailed.

| QMSum | Relevance | Correctness | Similarity | Faithfulness |
|---|---|---|---|---|
| Mistral7B | 51.82 | 8.97 | 52.90 | 69.39 |
| SARA | 70.37 | 11.17 | 53.51 | 70.68 |
| **MultifieldQA-en** | **Relevance** | **Correctness** | **Similarity** | **Faithfulness** |
| RAG | 42.32 | 21.97 | 42.09 | 31.61 |
| SARA | 90.04 | 22.24 | 45.13 | 32.56 |
| **2WikiMultiHopQA** | **Relevance** | **Correctness** | **Similarity** | **Faithfulness** |
| RAG | 31.50 | 35.69 | 29.91 | 42.82 |
| SARA | 86.53 | 37.87 | 31.58 | 44.13 |

Table 6: Results on out-of-domain datasets. We report Response Relevance (Relevance), Answer Correctness (Correctness), Semantic Similarity (Similarity), and Faithfulness (Faithfulness).

the retriever $\mathcal{R}$ and compressor $f$, GPU memory consumption remains stable, slightly up from 26.4 GB to 27.4 GB

**Efficiency Advantage over Compression-Based Baselines.** By design, SARA employs a bounded number of refinement iterations that scale linearly with $k$, preventing exponential slow down. In contrast, compression-based baselines such as CompAct (Yoon et al., 2024) incurs a larger computational burden, with TTFT exceeding 200 seconds and a generation throughput of 1.6 tokens per second. This demonstrates that SARA delivers substantial accuracy gains while maintaining fast inference speed and achieving context compression.

## C Discussion

**Extension to New Decoders** SARA is designed to be *model-agnostic*. All components–retriever, compressor, and the QA model–can be replaced with minimal effort. Note that the same decoder must be used across both *Compression Learning* (Section 2.3), *Instruction-tuning*, and *Generation* (Section 2.4). This is because the model learns to *interpret* compression vectors through its own decoder weights.

**Broader Applications.** Beyond answer generation, the framework could naturally extend to other retrieval-oriented tasks—such as document reranking, evidence attribution, or citation resolution—particularly in high-stakes domains (e.g., legal, scientific, or financial) where concise yet faithful representation is crucial.

## D Expressivity of compression vectors

Faithful representation of semantics is pivotal for our compression vectors to serve as reliable contexts. To evaluate this, we decode the compression vectors into natural language and compare the reconstructed evidence with their sources. Representative successes for both chunk-level and paragraph-level reconstructions are shown in Table 10 and 9. We observed that the decoded text are usually shorter and serve as higher level summarizations for the input. In most cases, the decoded text preserves core propositions, causal links, and sentiment. SARA is able to recover key information, such as exact entities (e.g. 'Amazon customer service') and numeric values (e.g. '220'). Losses are mostly fine-grained–exact dates ('1903' $\rightarrow$ '1900s') or numeric magnitudes ('3400 years' $\rightarrow$ over 3,000 years) may be paraphrased or omitted. When contexts are longer, the risk of recovery failure is higher. This necessitates reasoning over mixed evidence formats.

Crucially, the decoder rarely invents new facts: missing detail is typically dropped rather than hallucinated. This behavior implies that the vectors encode stable, high-level meaning while suppressing fewer specifics–a valuable feature for knowledge-intensive tasks that demand both factual precision and robust hallucination control.

## E Ethical Considerations

When deploying retrieval-augmented generation systems like SARA in sensitive domains such as legal or scientific applications, practitioners should exercise caution when interpreting compressed selected evidence. Human experts should remain

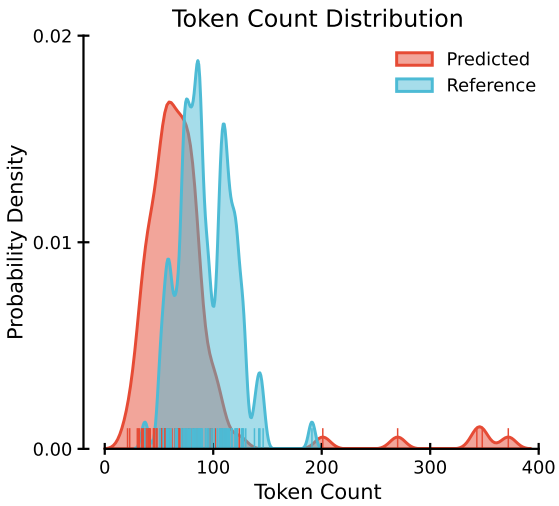

Figure 9: Probability density of token counts for decoded evidence and reference evidence with a 3-sentence generation budget.

in the loop to validate retrieved content and assess whether critical context has been preserved in the compression vectors. Transparency about compression mechanisms and retrieval limitations is essential to ensure responsible use, particularly in high-stakes settings.

## F Use of AI Assistants

AI assistants were used in a limited capacity to help refine the presentation of this paper, including language polishing, clarity, and organization. They also assisted with parts of data cleaning and processing. Experimental design, technical analysis, and final content decisions were verified by the authors.

| |
|---|
| **[Embedding Alignment]**<br><C> means the same as: <Sentence> |
| **[Context Reconstruction]**<br>Interpret the following tokens as a single document: <C> <C> ...<C>: <Paragraph> |
| **[Instruction-tuning / Inference]**<br>Using the context and additional context, answer the following question: <question><br>**Context**: <context><br>**Additional Context**:<br>1. <C>, <C>, ..., <C>;<br>2. <C>, <C>, ..., <C>;<br>**Question**: <Question><br>**Your Answer**: <Answer><br>**Judgment**: |

Table 7: Prompt for pretraining, instruction-tuning, and inference. <C> indicate positions for the compression vectors

| Question | How do they define rumors? |
|---|---|
| xRAG | Rumors are false, unverified information that spreads quickly. |
| LLMLingua | Rumors are not explicitly defined in the context provided. |
| LongLLMLingua | They define rumors as unverified information spread on social media. |
| **SARA** | They define rumors as unverified information that spreads rapidly through a population. |
| *Ground-truth* | They define rumors as the presence of rapidly spreading information that is unconfirmed by official media. |
| Question | What are the five downstream tasks? |
| xRAG | 1. Named Entity Recognition (NER)<br>2. Dependency Parsing (DP)<br>3. Sentiment Analysis (SA)<br>4. Machine Translation (MT)<br>5. Question Answering (QA) |
| LLMLingua | The five downstream tasks mentioned in the context are: machine translation, question answering, text summarization, sentiment analysis, and named entity recognition. |
| LongLLMLingua | The five downstream tasks related to grammatical error correction are text summarization, sentiment analysis, named entity recognition, question answering, and text classification. |
| **SARA** | The five downstream tasks are text / intent classification, NLI, named entity recognition, and POS tagging. |
| *Ground-truth* | The five downstream tasks we perform using MMTE are three classification tasks: NLI (XNLI dataset), document classification (MLDoc dataset), and intent classification, as well as two sequence tagging tasks: POS tagging and NER. |

Table 8: Comparison of answers generated by different compression methods.

| Prediction | Ground-truth |
|---|---|
| **# Anti-scam dataset**<br>Collecting **human-human conversational data** to create a dataset for training and evaluating **anti-scam** models. We **collect conversations** between **users** and **attackers who aim to gather customer information** from **Amazon customer service scam scenarios**. We collected 220 anti-scam conversational data from **Amazon customers** through a Turkers' platform, which are **human-human dialogues**. The average length of a conversation is 11.5 turns and the average length is 11 words. **172 out of 220 users successfully identified attackers**, indicating that **the attackers are well-trained** in their scam attack strategy. We recruited **two experienced annotators** to evaluate the quality of the annotated data. | **## AntiScam Dataset**<br>To enrich available non-collaborative task datasets, we created a corpus of **human-human anti-scam dialogs** in order to learn human elicitation strategies. We chose a popular **Amazon customer service scam scenario** to collect dialogs between users and attackers who aim to collect users information. We posted a role-playing task on the Amazon Mechanical Turk platform and collected a typing conversation dataset named AntiScam. We collected 220 human-human dialogs. The average conversation length is 12.45 turns and the average utterance length is 11.13 words. **Only 172 out of 220 users successfully identified their partner as an attacker**, suggesting that **the attackers are well trained** and not too easily identifiable. We recruited **two expert annotators** who have linguistic training to annotate 3,044 sentences in 100 dialogs, achieving a 0.874 averaged weighted kappa value. |
| **Exploration of oil in Nigeria began around 1900**, when oil was discovered in commercial quantities in the Niger Delta region. However, **large-quantities** was only **discovered later** in **1956** in **Oloibiri**. | Although the history of **oil exploration in Nigeria dates back to 1903**, non-commercial quantities of oil were not discovered there until 1953. **Commercial amounts of crude oil** were **later discovered** in **Oloibiri, Nigeria** in **1956**. |
| **The Great Trek** was a series of **migrations** of **Dutch-speaking settlers** from **Cape Colony in South Africa**, which **began in 1836** and lasted for several years. | **The Great Trek** was an eastward **migration** of **Dutch-speaking settlers** who travelled by wagon trains from **the Cape Colony** into the interior of modern **South Africa from 1836 onwards**. The exploratory treks, however, arrived at the bay of Port Natal in February 1835. |
| **The history of music** is **the study of music and its development over time**, from **prehistoric times** to **the present day**. The oldest known written music is the song "Hymn to the Sun" from the Sumerian civilization, which is believed to be **over 3,000 years** old. | **The history of music** covers **the historical development and evolution of music** from **prehistoric times** to **present day**. The "**oldest known song**" was written in cuneiform, dating to 3400 years ago from **Ugarit in Syria**. The first piece of unwritten music was made prior to the Paleolithic age 3.3 million years ago. |

Table 9: Reconstruction quality of compression tokens in SARA. Source-aligned spans are shown in **bold** and errors are underlined. SARA faithfully reproduces most original semantics with only minor hallucinations.

| Decoded Text | Original Text |
|---|---|
| We release the code and the data. | We release the code and data. |
| Also, we build a persuasive dialogue system to persuade people to donate to charity. | Furthermore, we also build a persuasion dialog system to persuade people to donate to charities. |
| Rigid templates limit creativity and diversity, resulting in loss of user engagement. | However, rigid templates lead to limited diversity, causing the user losing engagement. |
| The generation model is good at producing diverse responses but lacks coherence. | On the other hand, language generation models can generate diverse responses but are bad at being coherent. |
| Collaborative end-to-end systems have been developed to a great extent for the goal to build a user-friendly system that enables participants to work together with the system to achieve a common goal. | Considerable progress has been made building end-to-end dialog systems for collaborative tasks in which users cooperate with the system to achieve a common goal. |
| We use a hierarchical annotation scheme. This generic annotation method can be applied to different tasks. | To handle social content, we introduce a hierarchical intent annotation scheme, which can be generalized to different non-collaborative dialog tasks. |

Table 10: Decoded text from compression vectors using `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023a) as the base model. Information omitted from one text but present in the other is underlined. Compared to the original, SARA retains concise semantics and excels at capturing high-level concepts. In some cases, it may lose fine-grained details such as specific entities and numerical values.

| Method | TTFT | Latency | Tokens/s | Memory (MB) | GPU Mem. (MB) |
|---|---|---|---|---|---|
| RAG (BM25) | $3.61 \pm 2.12$ | $1.68 \pm 0.51$ | $11.04 \pm 3.12$ | $2183.01 \pm 8.60$ | $15457.98 \pm 32.84$ |
| RAG (SFR) | $6.14 \pm 3.47$ | $1.70 \pm 0.46$ | $10.41 \pm 2.34$ | $2232.58 \pm 8.12$ | $26442.18 \pm 86.93$ |
| SARA ($k$=1) | $19.36 \pm 3.62$ | $1.72 \pm 0.63$ | $10.10 \pm 2.46$ | $2546.02 \pm 10.31$ | $27459.14 \pm 44.43$ |
| SARA ($k$=5) | $19.91 \pm 0.86$ | $1.98 \pm 0.72$ | $7.13 \pm 1.21$ | $2741.58 \pm 13.79$ | $27460.18 \pm 44.43$ |
| CompAct | $203.34 \pm 232.70$ | $120.21 \pm 145.03$ | $1.62 \pm 0.71$ | $2337.36 \pm 12.62$ | $15465.21 \pm 60.08$ |

Table 11: Inference efficiency comparison across methods. For RAG, we experiment with retriever of BM25 (Robertson et al., 2004) and SFR (Meng et al., 2024). SARA uses $n = 10$ and varying $k$. TTFT: time-to-first-token (s). Latency: average response time per query (s). Tokens/s: generation throughput (tokens per second). Memory: peak CPU RAM usage. GPU Mem.: peak GPU memory usage. Dashes indicate metrics that are not directly measurable for hosted-API endpoints.

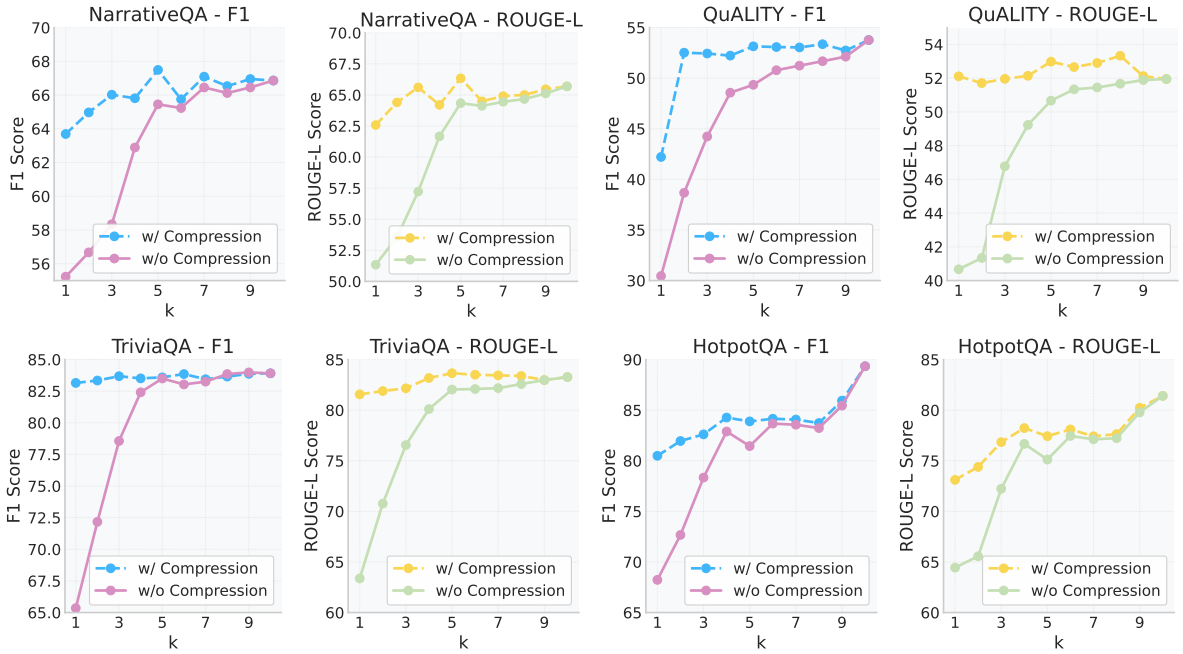


Figure 10: Sensitivity analysis with total contexts fixed at $N = 10$, varying the number of natural language contexts $k$ and effective compression ratios.