# Oreo: A Plug-in Context Reconstructor to Enhance Retrieval-Augmented Generation

Sha Li
Virginia Tech
Blacksburg, VA, USA
shal@vt.edu

Naren Ramakrishnan
Virginia Tech
Alexandria, VA, USA
naren@cs.vt.edu

## Abstract

Retrieval-Augmented Generation (RAG) aims to augment the capabilities of Large Language Models (LLMs) by retrieving and incorporating external documents or chunks prior to generation. However, even improved retriever relevance can bring erroneous or contextually distracting information, undermining the effectiveness of RAG in downstream tasks. We introduce a compact, efficient, and pluggable module designed to refine retrieved chunks before using them for generation. The module aims to extract and reorganize the most relevant and supportive information into a concise, query-specific, format. Through a three-stage training paradigm—comprising supervised fine-tuning, contrastive multi-task learning, and reinforcement learning-based alignment—it prioritizes critical knowledge and aligns it with the generator's preferences. This approach enables LLMs to produce outputs that are more accurate, reliable, and contextually appropriate.

## CCS Concepts

• **Information systems** → **Language models**.

## Keywords

Retrieval Augmented Generation, Prompt Optimization, Contrastive Learning

**ACM Reference Format:**

Sha Li and Naren Ramakrishnan. 2025. Oreo: A Plug-in Context Reconstructor to Enhance Retrieval-Augmented Generation . In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR) (ICTIR '25), July 18, 2025, Padua, Italy*. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3731120.3744590

## 1 Introduction

Large language models (LLMs) have demonstrated remarkable versatility across a wide spectrum of natural language processing (NLP) tasks, subsuming pipelines that were originally tailormade for each task. Despite being trained on massive text corpora, LLMs still face memory-related challenges such as out-of-date and out-of-domain knowledge, and they occasionally hallucinate non-factual or non-sensical content [64, 114]. To enhance the accuracy and reliability of LLM-generated outputs, retrieval-augmented generation (RAG) has emerged as a promising solution for knowledge-intensive tasks [26, 50, 115] (*e.g.* open-domain question and answering). RAG systems typically follow a *"retrieve-then-generate"* paradigm [77], where a *retriever* identifies relevant information from an external corpus and uses this information to augment context in constituting the input to a generative model (*i.e.*, *generator*), thus yielding an improved answer.

Despite its promise, a vanilla RAG system usually comes with shortcomings that can hinder its effectiveness. One major issue is semantic dissonance between the user query, the retriever, and the generator. This occurs when the retrieved documents, while semantically or contextually related to the topic, fail to directly address the query, leading to suboptimal answers [18, 94]. Another challenge pertains to the presence of noise, *i.e.* misleading, redundant, distracting, or even erroneous information within the retrieved documents. Such noise can misguide the generator, resulting in inaccurate or incoherent answers [78, 81]. For complex tasks that necessitate reasoning across multiple documents, the generator often struggles to correlate dependencies and relationships between them [8], leading to reasoning errors. For example, as illustrated in Figure 1, the correct answer while present in the retrieved chunks is not captured by the vanilla RAG process. Additionally, RAG systems are prone to the "lost-in-the-middle"[58] dilemma, where LLMs exhibit a tendency to prioritize information presented at the beginning and end of an input sequence, while paying less attention to the middle. Finally, the lack of joint optimization between the retriever and the generator exacerbates issues such as *knowledge inconsistencies* [85] or *knowledge conflicts* [97], which prevent the generator from producing accurate and contextually appropriate responses as the retrieved knowledge fails to adequately support the generation.

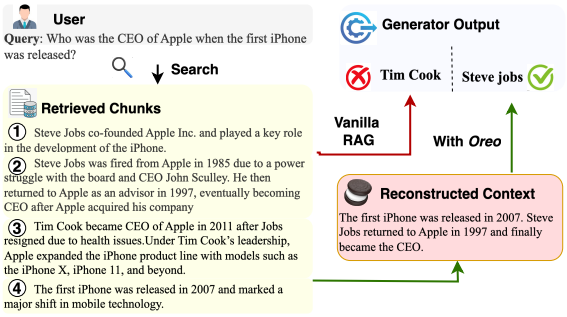


**Figure 1:** An example comparing vanilla RAG versus RAG with ***Oreo*** highlights the impact of redundant and scattered information within the retrieved document chunks. In the vanilla RAG setup, even though the retrieved chunks contain contextually relevant information to the query, the presence of distractions and redundancy misleads the downstream LLM, causing it to misinterpret temporal dependencies and generate an incorrect answer. In contrast, ***Oreo*** effectively captures the essential evidence and reconstructs the context, leading to accurate and correct responses.


*ICTIR '25, July 18, 2025, Padua, Italy*

ACM ISBN 979-8-4007-1861-8/2025/07
https://doi.org/10.1145/3731120.3744590

To address these challenges, many solutions have been proposed in prior research. Techniques such as query decomposition [13, 44], query rewriting [13, 60, 83, 87], and query expansion [48] aim to improve retriever performance by refining or enriching the input queries. Some studies have integrated rerankers [65, 104, 106] into retrieval systems, which reorder and prioritize the most relevant documents to ensure that the most pertinent information is provided to the generator. These works attempt to optimize the context on the passage level and largely ensure relevance with the query, but they still face challenges in maintaining comprehensive attention to the nuanced, finer-grained details of query-specific information.

Further advancements have been made in noise and redundancy exclusion. For example, filters based on lexical and information-theoretic approaches have been developed to identify and preserve useful content while directly eliminating less relevant information [38, 53, 88]. Summarization techniques [96] have been developed to synthesize and condense query-focused information from retrieved documents, leveraging extraction or abstraction methods. Compression techniques [12, 14, 15, 37, 38, 67, 103] extend this functionality by generating summary vectors that encode essential information for downstream tasks. While these methods improve efficiency, they do not align the retriever and generator in a manner that guarantees effective collaboration, which often result in knowledge gaps and consequently incorrect or suboptimal generation. From a training perspective, concurrent [30, 36, 55, 108] or asynchronous [79, 111] training of retrievers and generators is a widely adopted strategy to improve their interaction and collaboration [9, 30]. Although such techniques foster synergistic improvements, they can be computationally expensive and often require large amounts of annotated data to achieve optimal results.

In this work we introduce ***Oreo*** , a cOntext REcOnstructor designed to enhance the performance of RAG systems on knowledge-intensive tasks by *optimizing the quality of context* and *mitigating knowledge inconsistencies*. ***Oreo*** is implemented in a plug-and-play manner, functioning as an intermediary module between the retriever and the generator. It receives document chunks from the retriever and produces refined context tailored for the generator. Instead of merely extracting critical tokens from the chunks, ***Oreo*** reorganizes them and generates condensed query-aware summaries. Additionally, ***Oreo*** synergizes the reconstructed context with the generator's behavior of knowledge acquisition, ultimately leading to more accurate and contextually relevant answers.

Our work addresses the ICTIR 2025 theme of **"LLMs + IR, what could possibly go wrong?"** The prevailing assumption in many LLM+IR systems is that improving retriever relevance is always beneficial for downstream generation. However, our findings challenge this intuition: we do not always obtain better answers, especially when noisy, redundant, or poorly aligned content distracts the generator. ***Oreo*** exposes this tension and directly intervenes—reorganizing retrieved chunks into a cleaner, query-specific format that more faithfully serves generation. While many in the LLM ecosystem have embraced prompt optimization and instruction tuning to improve generation, relatively little attention has been paid to what happens in the in-between space between retrieval and generation (other than re-ranking). Our work brings this neglected middle into focus and thus core concerns that lie at the heart of the ICTIR 2025 theme.

Our key contributions are:

(1) We propose enhancing the RAG by introducing a "retrieve-reconstruct-then-generate" paradigm, offering a novel perspective on refining retrieved content for improved integration of external knowledge in RAG. ***Oreo*** overcomes the lack of contextual integration among fragmented chunks in vanilla RAG by extracting subtle relations from scattered facts, and transforming redundant context into a concise context.
(2) ***Oreo*** is a plug-and-play module, inherently modular, generalizable, flexible and robust, powered by a three-stage training scheme comprising supervised fine-tuning, contrastive multi-task learning and reinforcement learning. This enables seamless integration with arbitrary retrievers, generators, and off-the-shelf RAG systems.
(3) We demonstrate ***Oreo*** 's efficiency, effectiveness and robustness for both single-hop QA tasks (PopQA [62], NaturalQuestion (NQ) [45], TriviaQA (TQA) [40]), and multi-hop QA tasks (HopotQA [102], 2WikiMultiHopQA [31]). On average, ***Oreo*** contributes 5.115% downstream performance while reducing the input token length for generator by 12.87x.

## 2 Methodology

We begin with a quick primer on general RAG and formulate our problem (§2.1), followed by an overview of the proposed method (§2.2) and details of each step (§2.3, §2.4, §2.5 and §2.6).

### 2.1 Problem Formulation

A typical RAG system comprises of two primary components that work in tandem: the retriever $\mathcal{R}$ identifies and retrieves top-$k$ document chunks $\mathcal{D} = \{d_1, d_2, ..., d_k\}$ from an external knowledge base based on their relevance to a given query $q$; the generator $\mathcal{G}$ then produces the final answer for $q$ by conditioning on the combination of $\mathcal{D}$ and query $q$, formally expressed as $y = \mathcal{G}(\mathcal{D}, q)$. However, the performance of such a general pipeline is compromised by reasons we indicated in §1. Therefore, we propose ***Oreo*** to reconstruct context by extracting the most supportive evidence from $\mathcal{D}$, and producing a concise, query-aware context $C$ that aligns with the knowledge acquisition mechanics and preference of $\mathcal{G}$. An ideal $C$ should be produced after ***Oreo*** identifies essential entities and facts from $\mathcal{D}$, establishes their relations, and retains only the necessary information for $\mathcal{G}$ to effectively answer $q$. This process goes beyond plain information extraction, as it involves organizing and synthesizing content into a coherent and query-specific context. Therefore, we formulate the context reconstruction task as (itself) a text generation problem.

### 2.2 Method Overview

Our method extends the standard RAG paradigm from "retrieve-then-generate" to "retrieve-reconstruct-then-generate". Specifically, we train a text generation model $\mathcal{M}_\theta$, parameterized by $\theta$ to map the retrieved documents $\mathcal{D}$ into a refined context $c$ that enables the downstream generator $\mathcal{G}$ to produce more accurate answers for an input query: $c = f_{\mathcal{M}_\theta}(\mathcal{D}, q)$. The training of $\mathcal{M}_\theta$ involves three stages: (1) $\mathcal{M}_\theta$ is trained to learn the transformation from original documents to refined context using annotated datasets (§2.4). (2) Self-generated samples are incorporated to enhance the model's

ability to recognize and correct its own errors, thereby improving robustness and generalization (§2.5). (3) The reconstructed context is aligned with the generator's knowledge acquisition process by incorporating feedback from $\mathcal{G}$(§2.6). However, obtaining an annotated dataset with refined context for SFT is challenging. Drawing inspiration from prior work [7], we replace human annotation with advanced LLMs to generate high-quality synthetic oracle training data (§2.3). An overview of the framework is depicted in Figure 2.

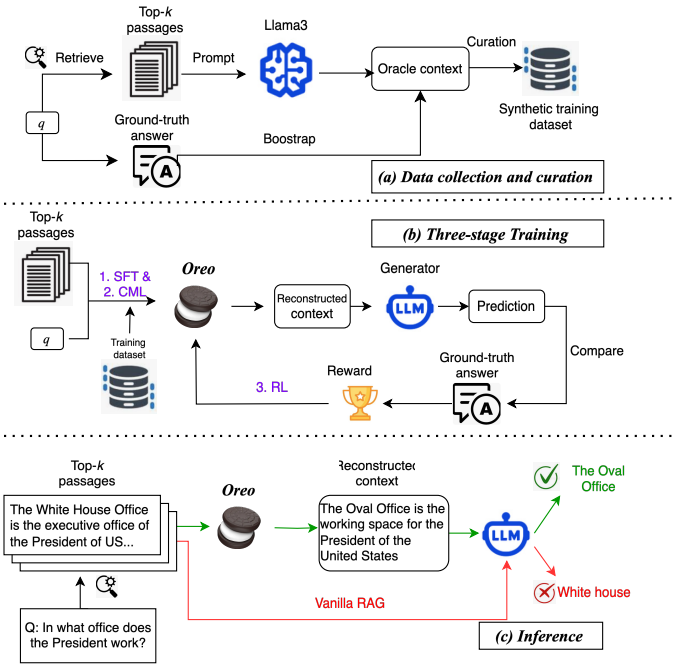


**Figure 2:** The framework of ***Oreo*** . (**top**) outlines the process of data collection and curation. (**middle**) demonstrates the three-stage training, which comprises the supervised fine-tuning (SFT), contrastive multi-task learning (CML) and reinforcement learning (RL) alignment. (**bottom**) illustrates the application of ***Oreo*** vs. vanilla RAG.

## 2.3 Data Collection and Curation

**Data collection**. To train ***Oreo*** during the SFT stage, an annotated dataset containing context with the most supportive evidence from retrieved document chunks is crucial. Such context should be query-specific, answer-aware, grounded in retrieved chunks, and structured as a rationale chain capable of deriving the correct answer. However, such datasets are not readily available, and manually annotating evidence for each query is time-consuming and labor-intensive. Fortunately, contemporary LLMs have exhibited impressive instruction learning capabilities to extract useful information [20] and generate high-quality reasoning steps [89] even in few-shot settings [10]. In this work, we elicit such capability from more advanced LLMs to our relatively smaller model ***Oreo*** , through generating a high-quality reasoning dataset using LLMs and utilizing it as "gold context" to train ***Oreo*** . Specifically, given a query and corresponding retrieved document chunks, we first prompt Llama3-8B-Instruct [84] to extract key entities and events from $\mathcal{D}$, and generate detailed rationales to answer the query. Since we prioritize the information extraction capability of ***Oreo*** during the SFT stage, to ensure reliability and minimize hallucinations, we construct the gold training dataset solely from query-document pairs where the ground-truth answer is explicitly present within the retrieved chunks.

**Bootstrapping**. For queries where the generated reasoning fails to include the ground truth (despite it being present in the retrieved chunks), we bootstrap Llama3 by providing the correct answer and iteratively reprompting it to perform generation. Such an iterative process allows Llama3 to reason backwards and learn to generate rationale chains that support the correct answer. This bootstrapping process is inspired by [90, 109]. The prompts and demonstrations used for gold context generation and boostrapping are provided in Appendix A.

**Data curation**. Accurate extraction of supporting evidence and reasoning from query to answer is essential for training $\mathcal{M}_\theta$. To eliminate hallucination and ensure the quality of learning, we conduct data curation by applying the following rules. *1. Ground Truth Alignment.* We retain query-context pairs where the generated context from Llama3 contains ground truth answers. *2. Entity and Event Consistency.* We extract sets of entities and events from both the original documents and the Llama3-generated context. Instances are retained only if the entities and events extracted from the generated context ($\mathcal{E}_{gen}$) are a subset of those present in the original documents ($\mathcal{E}_{ori}$). By following these steps, the refined context generated by Llama-3 is treated as "gold context" for training $\mathcal{M}_\theta$.

## 2.4 Supervised Fine-tuning

With the curated dataset constructed in §2.3, we employ supervised fine-tuning (SFT) to elicit the ability of extracting and reasoning from LLM to ***Oreo*** . Specifically, given a curated training dataset $\mathcal{T} = \{x_i, c_i\}_{i=1}^{N}$, where $x_i$ is the combination of query $q_i$, the associated retrieved document chunks $\mathcal{D}_i$ and task instructions. The goal of SFT is to train a sequential model $\mathcal{M}_\theta$ to generate target context conditioned on the $x$, and preceding tokens $c_{<t}$ of the context. The model minimizes the negative log-likelihood over the gold context, as defined by the following loss function:

$$\begin{aligned}\mathcal{L}_{SFT} &= \mathbb{E}_{(x,c)\sim\mathcal{T}}[-log\ p_{\mathcal{M}_\theta}(c|x)] \\ &= -\sum_{t=1}^{L} log\ p_{\mathcal{M}_\theta}(c_t|x, c_{<t})\end{aligned} \tag{1}$$

where $p$ represents probability distribution of generation by the model $\mathcal{M}_\theta$.

## 2.5 Contrastive Multitask Learning

The SFT in §2.4 serves as the initial step in equipping ***Oreo*** with the capability to reconstruct context. By emulating the behavior of an LLM, SFT enables ***Oreo*** to extract critical entities, events, and facts from $\mathcal{D}$, capture subtle relationships and organize them into coherent reasoning paths. This process ensures that the reconstructed context effectively supports the generation of accurate and complete answers for queries. However, autoregressive models trained solely on ground truth data often demonstrate suboptimal generalization performance. To address this issue, our goal is broader: we seek to empower ***Oreo*** to identify its own errors and integrate sequence-level supervised signals, which are critical for enhancing conditional text generation into training, thus improving its generalization. To achieve this goal, we introduce contrastive learning as a complementary step following SFT.

**Construct contrastive samples.** Inspired by [3], in addition to using in-batch instances, we gather contrastive samples from ***Oreo*** 's own predictions. Specifically, we obtain the model's top-$n$ recent predictions via beam search, rank and label them as positive and negative pairs $(c^+, c^-)$ in descending order of sequence-level similarity with the gold context $C$, using the ROUGE metric to measure the similarity.

**Margin-based pairwise loss.** To guide the learning process, we employ a pairwise margin-based loss that encourages ***Oreo*** to bring positive candidates closer semantically to the retrieved document chunks $\mathcal{D}$ while distancing negative ones. This ensures that the positive candidates generated by ***Oreo*** capture the essential and grounded information from $\mathcal{D}$ with the guidance of gold context, while discarding irrelevant information. The pairwise loss function is combined with the negative log-likelihood loss from SFT, forming a multi-task learning process. The final loss function is expressed as:

$$\begin{aligned}\mathcal{L}_{CL} = \sum\ & max\{0, cos(E_{\mathcal{D}}, E_c^-) - cos(E_{\mathcal{D}}, E_c^+) \\ & + \eta * (rank_{c^-} - rank_{c^+})\} \\ & + \alpha \mathcal{L}_{SFT}\end{aligned} \tag{2}$$

where $E$ denotes the vector representations, $\eta$ is the hyperparameter and $rank_{c^+/c^-}$ denotes the ranking position of the candidates respectively, meaning that the contrastive pair with a larger ranking gap should have a larger margin [3, 113].

## 2.6 Reinforcement Learning Alignment

The supervised fine-tuning and contrastive multitask learning stages equip ***Oreo*** with the ability to capture critical evidence and retain supportive information from retrieved content. However, knowledge inconsistencies among the retriever $\mathcal{R}$, ***Oreo*** and the generator $\mathcal{G}$ persist due to their independent optimization processes. Additionally, training ***Oreo*** with keeping $\mathcal{G}$ as a black-box precludes gradient back-propagation from $\mathcal{G}$ to update ***Oreo*** . To address these challenges, we incorporate reinforcement learning (RL) into ***Oreo*** 's training pipeline following the above training stages. This step enables ***Oreo*** learn from labeled ground truth of downstream tasks by aligning their output with the needs of $\mathcal{G}$ to produce correct answers. Specifically, we model $\mathcal{G}$ as a reward model and leverage the discrepancy between $\mathcal{G}$'s generated output and ground truth as reward signals. Proximal Policy Optimization (PPO) [66, 80] is employed to optimize ***Oreo*** in this alignment stage for its flexibility and alignment with our reward-based objective, which directly incorporates generator feedback without needing labeled or synthetic preference pairs.

**Policy formulation and optimization**. In this step, $\mathcal{M}_\theta$ serves as the policy $\pi_\theta$. It takes the reconstructed context $\hat{c}$ from prior training steps and returns a new $\hat{c}^{'}$, optimized by feedback from $\mathcal{G}$. The action space consists of all tokens in the corpus. At each step, the parameterized policy $\pi_{\theta_t}$ selects an action $a_t$ in a given state $s_t$ to maximize the discounted long-term reward $\mathbb{E}_{\pi_{\theta_t}}[\sum_{t-0}^{T} \gamma^t \mathcal{R}(s_t, a_t)]$. Specifically, the action $a_t$ is predicting the next token, and state $s_t$ is the sequence of all preceding tokens. The objective function is:

$$\begin{aligned}\mathcal{L}_{RL} = & \mathbb{E}[min(r_t(\theta) \cdot A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) \cdot A_t)] \\ & - \beta(V(s_t) - R_t)^2\end{aligned} \tag{3}$$

where $r_\theta = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ is the ratio of the updated policy $\pi_\theta$ to previous policy $\pi_{\theta_{old}}$. PPO ensures stable and efficient updates by clipping policy ratios, preventing excessively large changes that could destabilize training. The parameter $\epsilon$ defines how much the new policy can deviate from the old policy. $A_t$ is the advantage function, measures whether or not the action is better or worse than the policy's old behavior, estimated using Generalized Advantage Estimation (GAE) [76]: $A_t = \sum_{l=0}^{L} (\gamma\lambda)^l (R_{t+l} + \gamma V(s_{t+l-1}) - V(s_{t+l}))$ where $\gamma$ and $\lambda$ are discount factors. $V(s_t)$ is a critic network estimating the value of state $s_t$. $R_t$ is the estimated reward at time $t$. $\beta(V(s_t) - R_t)^2$ weighted by $\beta$ minimizes the discrepency between estimated and true values.

**Reward estimation**. With the downstream generator $\mathcal{G}$ serving as a reward model, the generation of ***Oreo*** by policy $\pi_{\theta_t}$ is passed to $\mathcal{G}$ with query $q$ to generate the answer $y$. When the end of sentence (*e.g.* <EOS>) token is generated, the corresponding reward $R_t$ is obtained by comparing the generated answer $y$ with ground truth answer $y_{gold}$, which is measured by the ROUGE score $R_t = ROUGE(y, y_{gold})$. However, $\mathcal{G}$ generates answers only after completing all tokens, but (3) updates every action step. To address this, we incorporate a token-level weighting mechanism [101]. Considering that a token $t$ with higher generation probability deemed more critical by the current policy. Consequently, the token's contribution to the final reward is proportionally adjusted. We estimate the reward at each step $t$ using the formulation:

$$R_t = \text{ROUGE}(y, y_{gold}) * \log(\text{softmax}(e^{\pi_\theta(a_t|s_t)})) \tag{4}$$

Since the rewards estimated by (4) are sequence-level and sparse, following [92], we regularize the reward function using a token-level KL penalty to prevent the model from deviating too far from the initialized LM. The final regularized reward estimation is:

$$\hat{R_t} = R_t - \delta KL(\pi_{\theta_t}(a_t|s_t) || \pi_0(a_t|s_t)) \tag{5}$$

# 3 Experiments

We evaluate ***Oreo*** across five open-domain question-answering (ODQA) tasks, comparing its performance against a suite of baselines. Our experiments holistically assess the quality of reconstructed context by ***Oreo*** along five critical dimensions: **efficiency**, **effectiveness**, **robustness**, **faithfulness** and **completeness**. Our primary emphasis is on **short-term factual QA tasks**, where the answers are typically concise in a few tokens. These tasks are sensitive to context quality and require precise evidence identification and summarization, making them an ideal benchmark for evaluating the performance of ***Oreo*** . In this section, we provide details of tasks and datasets (§3.1), baselines (§3.2) and experiment setup (§3.3).

## 3.1 Datasets and Tasks

**Datasets.** We evaluate ***Oreo*** on both single-hop and multi-hop open-domain question answering tasks. For single-hop QA, we use PopQA (PQA) [62], NaturalQuestions (NQ) [45], and TriviaQA (TQA) [40]. For multi-hop QA, we test ***Oreo*** on the more complex HopotQA (HQA) [102] and 2WikiMultiHopQA (2WQA) [31], where each question requires reasoning over multiple articles.

**External knowledge source.** For all experiments, we use the Wikipedia dump [41] as the external knowledge source.

**Evaluation metrics.** Following previous studies, *e.g.*, [88], we assess extractive QA performance (PopQA, NQ, and TriviaQA) using the Exact Match (EM) metric, while abstractive QA performance (HotpotQA and 2WQA) are measured using unigram F1.

We provide detailed statistics and experimental setups for each dataset in Appendix B.

## 3.2 Baselines

For comparison, we focus on evaluating how effectively ***Oreo*** enhances vanilla RAG systems treating both the retriever and generator as black-box components, acknowledging that they may be imperfect and not allowed to be fine-tuned. We compare the performance of downstream tasks using five configurations:

(1) **Query only**. The answer generation is performed by using only the query without incorporating any retrieved context. This mostly relies on the internal knowledge of LMs
(2) **Original full content**. The context for answer generation is the sequential concatenation of all retrieved document chunks. This setup uses raw, unprocessed retrieval results
(3) **Passage-level filtering.** Only the most relevant chunks are selected as context. Specifically, the chunk that is best-ranked is chosen for each query. For single-hop tasks, only one passage is selected, while for multi-hop tasks, two passages are used
(4) **Extraction and compression**. We employ eight state-of-the-art information extraction and compression methods to select informative sentences and generate concise summaries from retrieved documents. Specifically:
   (a) CXMI-trained model, following [88], uses conditional cross-mutual information (CXMI) [24] to train a language model to filter redundant context by quantifying each sentence's contribution to the correct answer
   (b) Selective-Context [52] removes uninformative content based on self-information
   (c) LLMLingua [37] and LLMLingua-2 [67] apply perplexity-based compression to retain sentences that most enhance answer likelihood
   (d) xRAG [14] encodes passages into a single embedding token, integrating them via modality fusion into the LM's representation space
   (e) CompAct [103] is a progressive compression framework that preserves query-aware content
   (f) EXIT [34] employs an adaptive extractive pipeline to select context based on query relevance
   (g) Refiner [54] uses a decoder-only LLM to extract verbatim query-relevant spans, organizing them by interdependencies
(5) **reconstructed context by using *Oreo***

## 3.3 Experiment setup

**Retriever**. To retrieve top-$k$ document chunks for each query ($k = 5$ unless otherwise specified), we employ a range of off-the-shelf retrievers, including Contriver [49], DPR [41] and BM25 [74]. The choice of multiple retrievers ensures that the robustness of ***Oreo*** is tested against various retrieval mechanisms, each with different strengths and weaknesses. Additionally, we extend our experiments to include retrieval of the top-*10* document chunks for 2WikiMultihopQA to examine whether ***Oreo*** 's performance is sensitive to context length.

**Downstream generator**. We access how do the contexts generated by different methods described in §3.2 affect the downstream generator by evaluating the performance of QA tasks. Specifically, we use FLAN-T5 [17] and OPT-IML [35] as the downstream generator. (Note that, ***Oreo*** operates as an independent module, making it compatible with various retrievers, generators, and other existing RAG frameworks. )

**Model and training**. We employ T5-small [72] as the backbone model for ***Oreo*** , though it is applicable to any encoder-decoder and auto-regressive models such as LLaMA. ***Oreo*** is implemented based on Transformer library [91], with RL implementation built upon the open-sourced package RL4LM [73]. For CML, we allow a maximum of 12 contrastive samples generated by ***Oreo*** and set beam size as 8. Unless otherwise specified, ***Oreo*** is trained for 5 epochs during SFT and 3 epochs for CML stages, with batch size 4/8/16 based on the dataset size, and using a learning rate of $5e-5$. Detailed parameter settings are listed in Appendix C.

**Inference**. During inference, we perform ablation studies by varying the number of tokens produced by ***Oreo*** to assess its impact on performance.

# 4 Results and Analysis

We seek to answer the following questions through experiments:

(1) How does ***Oreo*** perform in the RAG pipeline for QA tasks compared to alternative context configurations outlined in §3.2? (§4.1)
(2) To what extent does ***Oreo*** reduce input token length and balance inference latency while improving QA performance? (§4.2)
(3) How robust is ***Oreo*** to noisy contexts and perturbations in chunk order? (§4.3)
(4) How well does ***Oreo*** generalize to out-of-distribution datasets unseen during training? (§4.4)
(5) How effective is ***Oreo*** in generating context that are complete to the query and faithful to the retrieved chunks? (§4.5)
(6) How does the number of tokens in the reconstructed context affect downstream QA performance? (§4.6)

## 4.1 Effectiveness Evaluation

**Overall Performance**. Table 1 reports the average performance across single-hop and multi-hop QA tasks using Flan-T5 and OPT-IML as downstream generators, with contexts obtained through various retrieval, extraction, and compression strategies. Across all settings, ***Oreo*** consistently outperforms other approaches, achieving the highest performance on both single-hop (Exact Match) and multi-hop (F1) tasks. Flan-T5 generally delivers superior performance compared to OPT-IML, likely due to its more advanced instruction tuning.

**Comparison against context configurations.** Figure 3 presents the performance of different setups across five datasets (with Flan-T5 as downstream generator): using query-only inputs (without retrieval), original full content, passage-level filtering, ***Oreo*** with and without RL. The results demonstrate that ***Oreo*** surpasses all

**Table 1:** Average performance of ***Oreo*** across five QA benchmarks using Flan-T5 and OPT-IML as downstream generators, compared with baseline methods. *SC* denotes Selective-Context. Performance on single-hop and multi-hop QA tasks is evaluated using Exact Match and Unigram F1, respectively. **Bold** values indicate the best results.

| Task | No Retrieval | Full | Passage | CXMI | SC | LLMLingua | LLMLingua-2 | xRAG | CompAct | EXIT | Refiner | ***Oreo*** (Ours) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Flan-T5 as the downstream generator* | | | | | | | | | | | | |
| Single-hop QA | 0.1088 | 0.3662 | 0.3394 | 0.4016 | 0.2713 | 0.3491 | 0.269 | 0.2863 | 0.3603 | 0.3487 | 0.3680 | **0.4451** |
| Multi-hop QA | 0.4485 | 0.5671 | 0.5398 | 0.603 | 0.5297 | 0.5745 | 0.5576 | 0.4828 | 0.5974 | 0.5923 | 0.5925 | **0.658** |
| *OPT-IML as the downstream generator* | | | | | | | | | | | | |
| Single-hop QA | 0.125 | 0.2300 | 0.2698 | 0.2714 | 0.1696 | 0.1726 | 0.2461 | 0.2297 | 0.3142 | 0.3075 | 0.3160 | **0.3616** |
| Multi-hop QA | 0.4416 | 0.334 | 0.5866 | 0.4626 | 0.346 | 0.4363 | 0.5501 | 0.4818 | 0.5865 | 0.5828 | 0.5834 | **0.6542** |

other configurations across five datasets using Flan-T5. For single-hop QA tasks, ***Oreo*** achieves notable improvements in EM scores, with gains of 8.8%, 23.1%, and 37.5% on the PopQA, NQ, and TriviaQA datasets compared with using original full context respectively. The relatively smaller improvement on PopQA can be attributed to the nature of its queries, which involve rare and long-tail entities. In the case of more complex multi-hop QA tasks, ***Oreo*** achieves F1 score improvements of 12.7% on HotpotQA and 19.8% on 2WQA. These improvements are comparatively less pronounced than those seen in single-hop tasks. This discrepancy likely stems from the increased task complexity inherent in multi-hop QA. The additional challenge of ensuring coherence in abstractive multi-hop reasoning from fragmented chunks underscores the potential for further optimization in ***Oreo*** 's handling of such tasks. The experiments conducted on the 2WQA dataset using top-*5* and top-*10* retrieved document chunks demonstrate ***Oreo*** 's flexibility in handling different input lengths. The improved performance with the top-*10* chunks arises from the increased likelihood of covering more passages that contain the ground-truth answer.

Overall, these results reveals ***Oreo*** 's capability to capture essential information and filter out distracting content from retrieved document chunks, leading to improved performance in downstream factual QA tasks. The modest improvements achieved with RL further emphasize its value in addressing knowledge inconsistencies between the retriever and generator.

**Comparison against baselines.** We compare the quality of generated context by ***Oreo*** against a suite of representative extraction and compression methods across five QA datasets. Table 2 summarizes the performance and token counts across five datasets by employing Flan-T5 as the downstream generator. From the table, it is evident that ***Oreo*** outperforms almost all selected methods across five datasets, with improvements ranging from 0.35% to 8.58% over the second-best methods. These gains are particularly pronounced in extractive, single-hop tasks such as NQ and PopQA, where concise yet precise evidence retrieval is paramount. On multi-hop tasks, ***Oreo*** remains competitive but shows relatively smaller gains, as these tasks require complex evidence chaining and reasoning to synthesize evidence scattered across multiple chunks. In addition to achieving superior performance, ***Oreo*** significantly reduces the context length provided to the downstream generator while maintaining or even enhancing task accuracy. We further validate these findings by evaluating with OPT-IML as the downstream generator (see Fig. 4). Consistent with results of using Flan-T5, ***Oreo*** leads to the highest performance across four datasets (except HotpotQA),

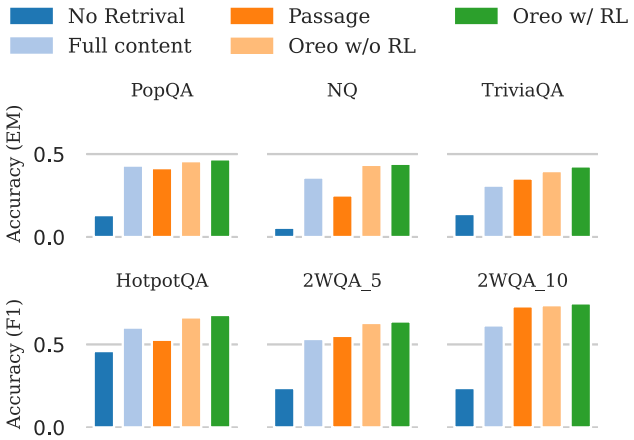


**Figure 3:** Performance on five datasets by using query without retrieval, original full concatenation of chunks, passage-level filtering, context generated by ***Oreo*** with and without RL. 2WQA_*k* represents retrieving top-*k* documents for the 2WQA dataset. The downstream generator is Flan-T5. Performance of PopQA, NQ and TriviaQA are measured by Exact Match and HotpotQA and 2WQA are measured by unigram F1.

bringing +0.0211 EM (PopQA), +0.0405 EM (NQ), +0.0019 EM (TriviaQA), -0.0142 F1 (HotpotQA) and +0.0495 F1 (2WQA) improvements compared with the second-best baseline method.

**SOTA methods observations.** Among selected SOTA methods, CompAct [103] marginally outperforms ***Oreo*** on HotpotQA, achieving a 2.26% higher F1 score. This advantage is attributed to CompAct's incremental and iterative compression strategy, which proves beneficial for tasks requiring deep multi-hop reasoning. However, its inference latency is nearly 3X that of ***Oreo*** , presenting a trade-off between accuracy and efficiency. Other strong performers include CXMI-guided model, Refiner and EXIT, which use query-aware or contrastive objectives to maintain relevance. In contrast, Selective-Context yields the weakest results overall. This underperformance likely stems from its reliance on self-information of lexical units (*e.g.*, tokens, phrases, or sentences), which fail to capture dependencies among semantic units. Similarly, LLMLingua and LLMLingua-2, which employ perplexity-based filtering, also struggle across datasets. Their reliance on self-information and perplexity metrics, without explicit query conditioning, limits their ability to extract context tightly aligned with user queries.

## 4.2 Efficiency Evaluation

We assess the efficiency of ***Oreo*** from two key perspectives: (1) the trade-off between context length reduction and downstream QA

**Table 2:** Summary of QA task performance and token counts using context derived from different methods. Flan-T5 is the generator. Performance on PopQA, NaturalQuestions, and TriviaQA is evaluated using Exact Match, while HotpotQA and 2WikiMultihopQA are assessed using Unigram F1. **Bold** values indicate the best performance among all methods, *italics* text denotes the second-best performance. The values in parentheses indicate the percentage improvement of the best-performing method over the second-best method. All datasets are tested with top-*5* retrieved chunks and all retrieved passages are set the same for different methods.

| Methods | PopQA | | NaturalQuestions | | TriviaQA | | HotPotQA | | 2WikiMultihopQA | |
|---|---|---|---|---|---|---|---|---|---|---|
| | EM | # tokens | EM | # tokens | EM | # tokens | F1 | # tokens | F1 | # tokens |
| No Retrieval | 0.1320 | 30 | 0.0558 | 39 | 0.1387 | 31 | 0.4599 | 47 | 0.4371 | 35 |
| Full content | 0.4305 | 1689 | 0.3584 | 1636 | 0.3097 | 1676 | 0.6014 | 1707 | 0.5328 | 1786 |
| CXMI | 0.4202 | 340 | 0.3917 | 329 | 0.3929 | 354 | 0.6409 | 351 | 0.565 | 305 |
| Selective Context | 0.1445 | 199 | 0.3981 | 193 | 0.2712 | 203 | 0.5588 | 214 | *0.6106* | 158 |
| LLMLingua | 0.2702 | 497 | *0.4125* | 491 | 0.3647 | 520 | 0.5584 | 527 | 0.5905 | 394 |
| Passage | 0.4150 | 131 | 0.2506 | 183 | 0.3526 | 203 | 0.5280 | 190 | 0.5515 | 205 |
| LLMLingua-2 | 0.2603 | 252 | 0.1892 | 247 | 0.3573 | 265 | 0.6186 | 269 | 0.4965 | 279 |
| xRAG | 0.2117 | 31 | 0.2580 | 40 | 0.2893 | 32 | 0.4777 | 48 | 0.4879 | 36 |
| CompAct | 0.3917 | 142 | 0.265 | 173 | *0.4242* | 180 | **0.6932** | 174 | 0.5015 | 173 |
| EXIT | 0.3972 | 124 | 0.2301 | 211 | 0.4188 | 208 | 0.6742 | 180 | 0.5104 | 123 |
| Refiner | *0.4312* | 91 | 0.2677 | 148 | 0.4051 | 148 | 0.6706 | 131 | 0.5144 | 102 |
| ***Oreo* (Ours)** | **0.4682 (+ 8.58%)** | **108** | **0.4413 (+6.98%)** | **134** | **0.4257 (+0.35%)** | **130** | *0.6775 (-2.26%)* | **272** | **0.6384 (+ 4.55%)** | **103** |

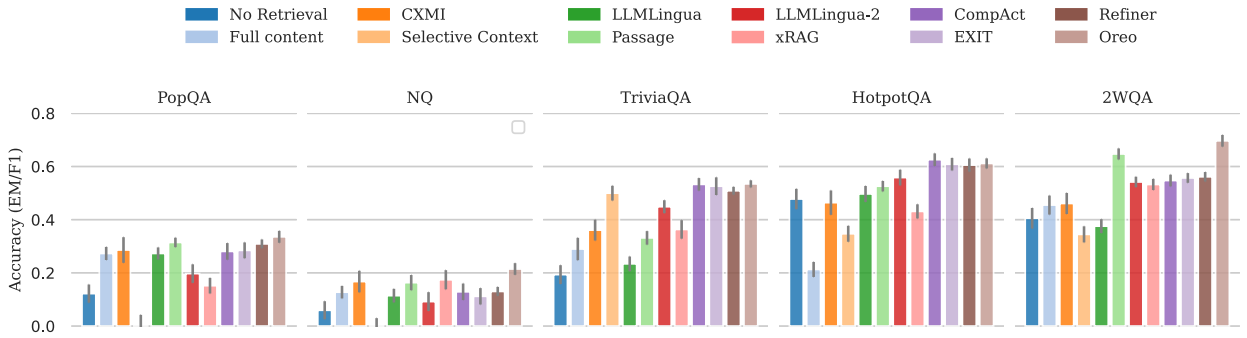


**Figure 4:** Performance comparison with 95% confidence intervals against baselines using OPT-IML as the generator. Specifically, *Passage* denotes passage-level filtering, *CXMI* refers to filtering guided by conditional cross-mutual information, and **Full** represents the use of original content without any filtering. PopQA, NQ, and TriviaQA are evaluated with Exact Match scores, while HotpotQA and 2WQA use Unigram F1 for accuracy measurement

performance, and (2) the trade-off between end-to-end inference latency and QA performance.

Figure 5 illustrates the number of tokens forwarded to the downstream generator and the total inference latency, which includes both ***Oreo*** 's context reconstruction and the subsequent generation time. We compare three input configurations: query-only (no retrieval), full document content, and the context reconstructed by ***Oreo*** . ***Oreo*** achieves a substantial reduction in input length, compressing the context by 84% to 94% compared to full document content. This compression is accompanied by a latency reduction of 22.98% to 43.01%, while simultaneously delivering significant performance improvements ranging from 8.76% to 37.46%. These gains are especially pronounced in extractive QA tasks (*e.g.* NQ, TriviaQA) as shown from the steeper improvement in Figure 5 from right endpoints to peaks. The high compression rate and improved performance demonstrates ***Oreo*** 's capability to effectively condense the retrieved context by preserving only the most critical evidence required for accurate answer generation. This also indicates the context reconstructed by ***Oreo*** is highly utilized by the downstream generator. The favorable trade-off between latency and performance underscores ***Oreo*** 's potential for real-world applications, offering both computational efficiency and improved task accuracy for scalable, high-throughput RAG systems.

## 4.3 Robustness Evaluation

We evaluate ***Oreo*** 's robustness from two aspects: its sensitivity to irrelevant or distracting information (noise robustness), and its ability to handle arbitrary rankings of retrieved chunks (order robustness).

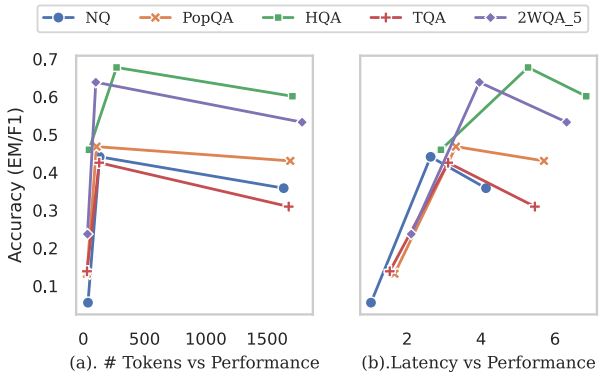


**Figure 5:** Left (a) - Comparison of number of input tokens for generator and QA performance across different context types. Right (b) - Comparison of end-to-end inference time (measured in seconds) by using different types of context.

**Noise robustness.** We evaluate the robustness of ***Oreo*** in handling noise within the retrieved documents, focusing specifically on extractive QA tasks. In this evaluation, we retain a single chunk that explicitly contains the ground-truth answer and introduce four irrelevant documents to simulate a noisy retrieval scenario. This setup examines ***Oreo*** 's effectiveness in filtering out distractions content and identifying query-specific information to generate accurate responses. Figure 6 depicts the performance degradation as irrelevant chunks are added to the context. Compared to directly concatenating all retrieved chunks as context, context reconstructed by ***Oreo*** demonstrates a smaller decrease in EM scores and a slower rate of decline, as evidenced by a less steep slope.

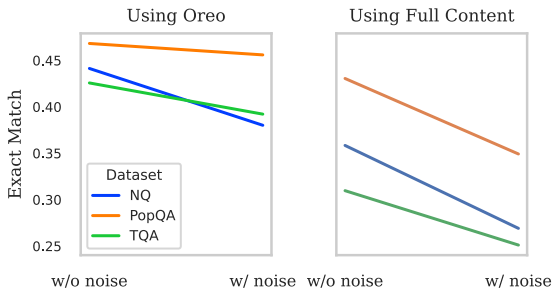


**Figure 6:** Performance declines as irrelevant chunks are introduced into the retrieved chunk set.

**Order robustness.** We evaluate the robustness of ***Oreo*** to variations in the order of retrieved documents by shuffling the top-*5* retrieved results and comparing its performance against the original document order. The results for five datasets are presented in Table 3. From the table we can see that, ***Oreo*** consistently maintain the performance on five datasets (with ±0.003 to ±0.027). This highlights that ***Oreo*** is order- or position-agnostic. Even the retrieved chunks are suboptimally ranked or presented in an arbitrary order, ***Oreo*** can still effectively capture and synthesize essential information as long as the evidence exists in the chunks. This capability is largely attributed to ***Oreo*** 's inherent reordering feature during context reconstruction, enabling it to function as an implicit reranker. Such robustness is particularly valuable for mitigating the "lost-in-the-middle" [58] phenomenon, where the order of relevant information may influence the downstream generator's performance.

**Table 3:** QA performance when shuffling the retrieved documents.

| Dataset | w/o shuffle | w/ shuffle |
|---|---|---|
| PopQA | 0.468 | 0.441 |
| NaturalQuestions | 0.441 | 0.425 |
| HotpotQA | 0.426 | 0.429 |
| TriviaQA | 0.678 | 0.668 |
| 2WikiMultihopQA | 0.638 | 0.614 |

## 4.4 Generalizability Evaluation

To evaluate the cross-dataset generalizability of ***Oreo*** , we assess its transferring capability by applying models trained on one dataset to tasks in a different dataset without any fine-tuning. This approach tests ***Oreo*** 's ability to generalize its context reconstruction and synthesis capabilities to unseen query distributions. Specifically, we examine performance when using a model trained on PopQA to generate answers for NQ and a model trained on 2WQA to process HotpotQA queries. We report the detailed results in Table 8 in Appendix E, which demonstrate that ***Oreo*** achieves competitive performance in the zero-shot setting. For example, the model trained on PopQA achieves a score of 0.4352 when applied to NQ, only slightly lower than the performance of being specific trained (*i.e.* 0.4413 and 0.4682). Similarly, using the 2WQA-trained model on HotpotQA yields a score of 0.6344, closely matching the intra-dataset score of 0.6384. These findings demonstrate ***Oreo*** 's ability to generalize its context reconstruction to similar QA types effectively, even under query distribution shifts. Its strong performance across datasets highlights its robustness and adaptability, making it a promising solution for open-domain QA tasks that require flexibility in handling diverse knowledge sources and query structures.

## 4.5 Faithfulness and Completeness Evaluation

Apart from the downstream task performance, the quality of context generated by ***Oreo*** is essential, esp. the factual accuracy (faithfulness) and coverage of relevant information (completeness) with respect to the original retrieved passages. To this end, we conduct an evaluation of both faithfulness and completeness to ensure that ***Oreo*** produces context that is not only concise but also reliable and fully representative of the source passages.

We adopt the LLM-as-a-judge framework [29] to systematically assess these dimensions. In particular, we prompt Qwen2.5-Instruct [100] to evaluate the generated context by assigning faithfulness and completeness scores on a 0–10 scale. *Faithfulness* reflects the degree to which the context remains factually grounded in the original retrieved content, avoiding hallucinations or the introduction of extraneous information. *Completeness* assesses whether the context sufficiently captures all salient and relevant details from the original passages with respect to the query. To promote transparency and interpretability, the model is also asked to provide a rationale supporting each score. Prompts for scores and explanation generation are provided in Appendix D.

We provide evaluation results in Figure 7 in Appendix D.2 Among all methods, ***Oreo*** consistently achieves the highest scores across all datasets, excelling in both completeness and faithfulness, with standout scores on complex datasets like HotpotQA (8.87 completeness, 8.97 faithfulness). CompAct emerges as a strong second-best, showing strong balance and high faithfulness, especially on HotpotQA and 2WQA. Refiner delivers moderately competitive results, generally maintaining factual consistency but showing limitations in coverage. EXIT demonstrates lower overall performance, especially struggling on more demanding datasets such as 2WQA. In contrast, LLMLingua and LLMLingua-2 produce the weakest results, with both completeness and faithfulness significantly lower across all datasets, likely due to aggressive filtering or compression strategies that sacrifice critical information.

## 4.6 Ablation Study

We conduct an ablation study to investigate the effect of varying context lengths generated by ***Oreo*** . Specifically, we progressively increase the minimum token threshold for context generation from 30 to 300 tokens, in increments of 30, while fixing the number of retrieved passages to top-*5*. The results of downstream task performance are summarized in Figure 8 in Appendix **??**. Our findings

indicate a consistent performance improvement across all datasets as the minimum context length increases, with gains being more pronounced in the earlier stages (from 30 to 180 tokens). These improvements suggest that extending the context allows the model to access a broader set of relevant evidence, improving its ability to synthesize accurate responses. However, beyond a threshold, typically between 240 and 270 tokens, we observe a performance plateau or marginal decline. This indicates diminishing returns with excessively long contexts. While longer windows can potentially capture more relevant details, they also risk introducing extraneous, redundant, or weakly relevant content, which can dilute the core information necessary for accurate answers.

# 5 Related Work

## 5.1 Post-retrieval Enhancement for RAG

Post-processing methods are widely employed to refine retrieved content for improved downstream generation. These methods can be categorized as follows:

**Reranking**. Rerankers reorder and prioritize retrieved documents to emphasize the most relevant results. They typically operate sequentially or iteratively after retrieval, leveraging various criteria such as semantic relevance between query and passages [28, 32], connections among documents [23], the majority of reader predictions [42, 63], and utility for generation [61]. Rerankers are usually based on cross-encoder (*e.g.* BGE [95], Mixedbread [51]), multi-vector models (*e.g.* ColBert [43, 75]). Recent works also explore using LMs as rerankers (*e.g.*, RankT5 [116], RankZephyr [69], RankGPT [82], DPA-RAG [22]).

**Post verification and correction**. Some studies incorporate post-hoc evaluations to improve factuality and relevance of retrieved documents. Examples include relevance evaluators [99], fact-checkers [56], attribution [25, 105] and multi-agent [98] mechanisms to further solidify the accuracy and reliability of the retrieved documents and responses.

**Compressing**. Compression methods condense retrieved content to improve efficiency and focus. These methods can be broadly categorized into lexical-based and embedding-based approaches. Lexical-based methods usually involve summarization techniques [57, 96] to retain essential information, semantic filtering to remove low-importance tokens, and both extractive and abstractive strategies for eliminating irrelevant context [96]. Some approaches compute the self-information of lexical units to discard less informative content [53], or apply token-level filtering based on perplexity [38]. Embedding-based methods, on the other hand, condense documents into fixed-size representations in the embedding space, recent works include xRAG [14] and PISCO [59]. Our work falls falls within the lexical-based compression group.

## 5.2 Reinforcement Learning for Large Language Models

Reinforcement learning for Language Models (RL4LM) has emerged as a transformative technique to further enhance LLMs' performance during the post-training process [11, 70]. Traditional RL4LM usually involves a reward model, for example using PPO [76] to update the policy model (*e.g.* InstructGPT [66], GPT-4 [1]). Some RL4LM such as Direct Preference Optimization (DPO) [71] and Reward-aware Preference Optimization (RPO) [2] get rid of the reward model to provide more stable and computationally efficient solutions (*e.g.* Qwen 2 [16] and Nemotron-4 [2]). Common goals of RL4LM include improving performance of downstream NLP tasks [21, 27, 73], minimizing data and resource dependencies [112], aligning model outputs with user intent, values and goals [66], and adhering to responsible AI principles [5, 6]. Human feedback can be integrated into the framework by constructing preference datasets, which are then used to fine-tune both the policy and reward models (also termed as Reinforcement Learning from Human Feedback (RLHF)) [5, 33, 66]. Some studies also explore RL4LM without human feedback [71] or replaced with AI feedback [6, 107] by distillation from LLMs [19, 68], prompting LLMs as reward functions [46, 47, 110], and self-rewarding [107], or using performance-based metrics such as fluency or coherence [27], and task-specific constraints over the distribution of language [73, 93]. In the specific domain of RAG, RRAML [4] employs RL to train a retriever in arbitrarily large databases. PRCA [101] applies RL to fine-tune the context to optimize the reward for the generator. BIDER [39] adopts RL to bridge the inconsistency between the retriever and generator.

# 6 Conclusion

We have presented ***Oreo*** - a lightweight and pluggable module designed to enhance the performance of RAG systems by reconstructing retrieved document chunks and mitigating the potential knowledge inconsistencies between the retriever and generator. ***Oreo*** can be seamlessly integrated with arbitrary retrievers, generators, or other RAG components without requiring significant adjustments or modifications. Experimental results demonstrate ***Oreo*** 's effectiveness in downstream tasks, its efficiency in compressing context while improving performance, and its robustness in handling noisy and imperfectly ranked document chunks.

**Limitations.** While ***Oreo*** shows strong performance on open-domain QA tasks, it has some limitations. First, its aggressive compression may omit essential information in complex settings like multi-hop or long-form QA. Second, ***Oreo*** has not been systematically tested in adversarial retrieval scenarios [86] involving conflicting or deceptive content. Third, current evaluations rely heavily on indirect metrics such as downstream QA accuracy or LLM judgments, which may introduce bias.

**Future Work.** Future research will explore adaptive compression strategies that dynamically allocate token budgets as well as robustness to adversarial or noisy retrieval scenarios. We are also interested in developing more principled and fine-grained evaluation frameworks to better understand the trade-offs between compression, informativeness and faithfulness in context reconstruction. To address sparsity in rewards, a promising direction for future work is to develop progress-based RL frameworks that incorporate intermediate quality assessments of ***Oreo*** 's reconstructed context, providing denser and more fine-grained rewards to enable more stable and efficient policy learning.

## Acknowledgments

This work is supported by the NSF under grant IIS-2312794 and the Amazon-Virginia Tech Initiative in Efficient and Robust Machine Learning. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

## References


[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340B Technical Report. *arXiv preprint arXiv:2406.11704* (2024).

[3] Chenxin An, Jiangtao Feng, Kai Lv, Lingpeng Kong, Xipeng Qiu, and Xuanjing Huang. 2022. Cont: Contrastive neural text generation. *Advances in Neural Information Processing Systems* 35 (2022), 2197–2210.

[4] Andrea Bacciu, Florin Cuconasu, Federico Siciliano, Fabrizio Silvestri, Nicola Tonellotto, and Giovanni Trappolini. 2023. RRAML: reinforced retrieval augmented machine learning. *arXiv preprint arXiv:2307.12798* (2023).

[5] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).

[6] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).

[7] Vidhisha Balachandran, Hannaneh Hajishirzi, William Cohen, and Yulia Tsvetkov. 2022. Correcting Diverse Factual Errors in Abstractive Summarization via Post-Editing and Language Model Infilling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 9818–9830. https://doi.org/10.18653/v1/2022.emnlp-main.667

[8] Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2022. Can retriever-augmented language models reason? the blame game between the retriever and the language model. *arXiv preprint arXiv:2212.09146* (2022).

[9] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[11] Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2024. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems* (2024).

[12] 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 12685–12695. https://doi.org/10.18653/v1/2024.acl-long.685

[13] Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. Rq-rag: Learning to refine queries for retrieval augmented generation. *arXiv preprint arXiv:2404.00610* (2024).

[14] 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* (2024).

[15] 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*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 3829–3846. https://doi.org/10.18653/v1/2023.emnlp-main.232

[16] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. 2024. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759* (2024).

[17] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[18] Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 719–729.

[19] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. (2023).

[20] John Dagdelen, Alexander Dunn, Sanghoon Lee, Nicholas Walker, Andrew S Rosen, Gerbrand Ceder, Kristin A Persson, and Anubhav Jain. 2024. Structured information extraction from scientific text with large language models. *Nature Communications* 15, 1 (2024), 1418.

[21] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (Eds.). Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 3369–3391. https://doi.org/10.18653/v1/2022.emnlp-main.222

[22] Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Zhicheng Dou, and Ji-Rong Wen. 2024. Understand what LLM needs: Dual preference alignment for retrieval-augmented generation. *arXiv preprint arXiv:2406.18676* (2024).

[23] Jialin Dong, Bahare Fatemi, Bryan Perozzi, Lin F Yang, and Anton Tsitsulin. 2024. Don't Forget to Connect! Improving RAG with Graph-based Reranking. *arXiv preprint arXiv:2405.18414* (2024).

[24] Patrick Fernandes, Kayo Yin, Graham Neubig, and André F. T. Martins. 2021. Measuring and Increasing Context Usage in Context-Aware Machine Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 6467–6478. https://doi.org/10.18653/v1/2021.acl-long.505

[25] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023. RARR: Researching and Revising What Language Models Say, Using Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 16477–16508. https://doi.org/10.18653/v1/2023.acl-long.910

[26] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[27] Demian Ghalandari, Chris Hokamp, and Georgiana Ifrim. 2022. Efficient Unsupervised Sentence Compression by Fine-tuning Transformers with Reinforcement Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1267–1280. https://doi.org/10.18653/v1/2022.acl-long.90

[28] Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, Rerank, Generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 2701–2715. https://doi.org/10.18653/v1/2022.naacl-main.194

[29] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594* (2024).

[30] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*. PMLR, 3929–3938.

[31] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

[32] Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1437–1447.

[33] Jian Hu, Li Tao, June Yang, and Chandler Zhou. 2023. Aligning language models with offline reinforcement learning from human feedback. *arXiv preprint arXiv:2308.12050* (2023).

[34] 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 preprint arXiv:2412.12559* (2024).

[35] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization. arXiv:2212.12017 [cs.CL]

[36] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard

Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299* 1, 2 (2022), 4.

[37] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 13358–13376. https://doi.org/10.18653/v1/2023.emnlp-main.825

[38] Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 1658–1677. https://doi.org/10.18653/v1/2024.acl-long.91

[39] Jiajie Jin, Yutao Zhu, Yujia Zhou, and Zhicheng Dou. 2024. BIDER: Bridging Knowledge Inconsistency for Efficient Retrieval-Augmented LLMs via Key Supporting Evidence. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 750–761. https://doi.org/10.18653/v1/2024.findings-acl.42

[40] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Vancouver, Canada.

[41] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

[42] Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the Preference Gap between Retrievers and LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 10438–10451. https://doi.org/10.18653/v1/2024.acl-long.562

[43] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[44] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. 2023. Tree of Clarifications: Answering Ambiguous Questions with Retrieval-Augmented Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 996–1009. https://doi.org/10.18653/v1/2023.emnlp-main.63

[45] 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 (2019), 453–466.

[46] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. 2023. Reward design with language models. *arXiv preprint arXiv:2303.00001* (2023).

[47] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. (2023).

[48] Yibin Lei, Yu Cao, Tianyi Zhou, Tao Shen, and Andrew Yates. 2024. Corpus-Steered Query Expansion with Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, St. Julian's, Malta, 393–401. https://aclanthology.org/2024.eacl-short.34

[49] Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised Dense Retrieval with Relevance-Aware Contrastive Pre-Training. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 10932–10940. https://doi.org/10.18653/v1/2023.findings-acl.695

[50] Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110* (2022).

[51] Xianming Li and Jing Li. 2023. AnglE-optimized Text Embeddings. *arXiv preprint arXiv:2309.12871* (2023).

[52] Yucheng Li. 2023. Unlocking context constraints of llms: Enhancing context efficiency of llms with self-information-based content filtering. *arXiv preprint arXiv:2304.12102* (2023).

[53] 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*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 6342–6353. https://doi.org/10.18653/v1/2023.emnlp-main.391

[54] Zhonghao Li, Xuming Hu, Aiwei Liu, Kening Zheng, Sirui Huang, and Hui Xiong. 2024. *Refiner*: Restructure Retrieved Content Efficiently to Advance Question-Answering Capabilities. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA. https://doi.org/10.18653/v1/2024.findings-emnlp.500

[55] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352* (2023).

[56] Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and J Wen. [n. d.]. RETA-LLM: a retrieval-augmented large language model toolkit (2023). *arXiv preprint arXiv:2306.05212* ([n. d.]).

[57] Junyi Liu, Liangzhi Li, Tong Xiang, Bowen Wang, and Yiming Qian. 2023. TCRA-LLM: Token Compression Retrieval Augmented Large Language Model for Inference Cost Reduction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9796–9810. https://doi.org/10.18653/v1/2023.findings-emnlp.655

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

[59] Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. PISCO: Pretty Simple Compression for Retrieval-Augmented Generation. *arXiv preprint arXiv:2501.16075* (2025).

[60] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query Rewriting in Retrieval-Augmented Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5303–5315. https://doi.org/10.18653/v1/2023.emnlp-main.322

[61] Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples!. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 10572–10601. https://doi.org/10.18653/v1/2023.findings-emnlp.710

[62] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 9802–9822. https://doi.org/10.18653/v1/2023.acl-long.546

[63] Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021. Reader-Guided Passage Reranking for Open-Domain Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 344–350. https://doi.org/10.18653/v1/2021.findings-acl.29

[64] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[65] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[66] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[67] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 963–981. https://doi.org/10.18653/v1/2024.findings-acl.57

[68] Junsoo Park, Seungyeon Jwa, Ren Meiying, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging Debiased Data for Tuning Evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 1043–1067. https://doi.org/10.18653/v1/2024.findings-emnlp.57

[69] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *arXiv preprint*

*arXiv:2312.02724* (2023).

[70] Moschoula Pternea, Prerna Singh, Abir Chakraborty, Yagna Oruganti, Mirco Milletari, Sayli Bapat, and Kebei Jiang. 2024. The RL/LLM Taxonomy Tree: Reviewing Synergies Between Reinforcement Learning and Large Language Models. *arXiv preprint arXiv:2402.01874* (2024).

[71] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).

[72] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[73] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[74] Stephen E Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. Springer, 232–241.

[75] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 3715–3734. https://doi.org/10.18653/v1/2022.naacl-main.272

[76] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015).

[77] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9248–9274. https://doi.org/10.18653/v1/2023.findings-emnlp.620

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

[79] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-Augmented Black-Box Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Kevin Duh, Helena Gomez, and Steven Bethard (Eds.). Association for Computational Linguistics, Mexico City, Mexico, 8371–8384. https://doi.org/10.18653/v1/2024.naacl-long.463

[80] Nisan Stiennon, Ouyang Long, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Francis Christiano. 2020. Learning to summarize with human feedback. In *Neural Information Processing Systems*. https://api.semanticscholar.org/CorpusID:263874153

[81] Weiwei Sun, Zhengliang Shi, Shen Gao, Pengjie Ren, Maarten de Rijke, and Zhaochun Ren. 2023. Contrastive learning reduces hallucination in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 13618–13626.

[82] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14918–14937. https://doi.org/10.18653/v1/2023.emnlp-main.923

[83] Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small Models, Big Insights: Leveraging Slim Proxy Models To Decide When and What to Retrieve for LLMs. *arXiv preprint arXiv:2402.12052* (2024).

[84] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[85] Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023. A Causal View of Entity Bias in (Large) Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 15173–15184. https://doi.org/10.18653/v1/2023.findings-emnlp.1013

[86] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Retrieval-Augmented Generation with Conflicting Evidence. *arXiv preprint arXiv:2504.13079* (2025).

[87] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

[88] 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* (2023).

[89] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[90] Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. InstructRAG: Instructing Retrieval-Augmented Generation with Explicit Denoising. *arXiv preprint arXiv:2406.13629* (2024).

[91] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

[92] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862* (2021).

[93] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024. $beta$-DPO: Direct Preference Optimization with Dynamic $beta$. *arXiv preprint arXiv:2407.08639* (2024).

[94] Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How Easily do Irrelevant Inputs Skew the Responses of Large Language Models? *arXiv preprint arXiv:2404.03302* (2024).

[95] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]

[96] Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2024. RECOMP: Improving Retrieval-Augmented LMs with Context Compression and Selective Augmentation. In *The 12th International Conference on Learning Representations*.

[97] Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge Conflicts for LLMs: A Survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 8541–8565. https://doi.org/10.18653/v1/2024.emnlp-main.486

[98] Zhipeng Xu, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Chaojun Xiao, Zhiyuan Liu, Ge Yu, and Chenyan Xiong. 2024. ActiveRAG: Autonomously Knowledge Assimilation and Accommodation through Retrieval-Augmented Agents. *arXiv preprint arXiv:2402.13547* (2024).

[99] Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884* (2024).

[100] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

[101] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. 2023. PRCA: Fitting Black-Box Large Language Models for Retrieval Question Answering via Pluggable Reward-Driven Contextual Adapter. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, 5364–5375. https://doi.org/10.18653/v1/2023.emnlp-main.326

[102] 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*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, Brussels, Belgium, 2369–2380. https://doi.org/10.18653/v1/D18-1259

[103] Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. 2024. CompAct: Compressing Retrieved Documents Actively for Question Answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 21424–21439. https://doi.org/10.18653/v1/2024.emnlp-main.1194

[104] Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558* (2023).

[105] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-Note: Enhancing Robustness in

Retrieval-Augmented Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 14672–14685. https://doi.org/10.18653/v1/2024.emnlp-main.813

[106] Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210* (2023).

[107] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020* (2024).

[108] Hamed Zamani and Michael Bendersky. 2024. Stochastic rag: End-to-end retrieval-augmented generation through expected utility maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2641–2646.

[109] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems* 35 (2022), 15476–15488.

[110] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240* (2024).

[111] LingXi Zhang, Yue Yu, Kuan Wang, and Chao Zhang. 2024. ARL2: Aligning Retrievers with Black-box Large Language Models via Self-guided Adaptive Relevance Labeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 3708–3719. https://doi.org/10.18653/v1/2024.acl-long.203

[112] Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. 2022. Tempera: Test-time prompting via reinforcement learning. *arXiv preprint arXiv:2211.11890* (2022).

[113] Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 6197–6208. https://doi.org/10.18653/v1/2020.acl-main.552

[114] Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting Hallucinated Content in Conditional Neural Sequence Generation. In *Findings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP Findings)*. Virtual. https://arxiv.org/abs/2011.02593

[115] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107* (2023).

[116] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2308–2313.

# Appendices

## A Prompt Templates for Data Collection

### A.1 Prompt Templates for Data Collection

**Input:** Your task is to decompose the question, extract and abstract supporting information from the context to answer the question. Your output should mention all entities involved in the question, supporting sentences and rationals to all sub-questions from the context. If the conetxt doesn't provide information to answer the question, output '[UNKNOWN]'. Output the <Output> part only.
Example1:
<Question>: Where was the director of film The Fascist born?
<Context>: {Retrieved document chunks}
<Output>: Luciano Salce, the director of the satirical film The Fascist, was born on September 25, 1922, in Rome, Italy. Salce was an Italian filmmaker, actor, and screenwriter known for his ability to blend comedy with social and political critique.
Example2:
<Question>: what is the number 1 sport in the usa?
<Context>: {Retrieved document chunks}
<Output>: American football is the most popular sport in the United States followed by basketball, baseball, and soccer.
Example3:
<Question>: What was the first English monastery to be sacked by the Norsemen?
<Context>: {Retrieved document chunks}
<Output>: Vikings attacked the monastery at Lindisfarne on June 8, 793, which is the first recorded Viking raid on an English monastery.
Example4:
<Question>: Kate Philips played which wife of Henry VIII in 'Wolf Hall'?
<Context>: {Retrieved document chunks}
<Output>: Kate Phillips played Abigail Williams in "The Crucible" at the West Yorkshire Playhouse, and then went on to film her scenes for the BBC's adaptation of "Wolf Hall" in which she played Jane Seymour, Henry VIII's third wife.
Example5:
<Question>: Lokomotiv Yaroslavl was the team founded in 2011 after the plane crash near which airport?
<Context>: {Retrieved document chunks}
<Output>: Lokomotiv Yaroslavl Hockey Club Lokomotiv, also known as Lokomotiv Yaroslavl, is a Russian professional ice hockey team. On 7 September 2011, nearly the entire team perished in the Lokomotiv Yaroslavl plane crash. The aircraft ran off the runway before lifting off, struck a tower mast, caught fire and crashed from the end of the runway of Tunoshna Airport on the Volga River bank.

{Question}
{Retrieved document chunks}
**Output:** {Output}

### A.2 Prompt Template for Boostraping Data Generation

**Input:** You are given a question, a set of document chunks, a correct answer, extract evidences and supporting information from the chunks and generate rationales how these information derive the correct answer.
Example1:
<Question>: What nationality were social anthropologists Alfred Gell and Edmund Leach?
<Chunks>: {Retrieved document chunks}
<Correct answer>: British.
<Output>: Both Alfred Gell and Edmund Leach were British. They were educated and primarily worked within the United Kingdom's academic framework. Their national and professional affiliations firmly establish their British nationality.
Example2:
<Question>: Crucible is a geodemography computer system created by a company that has stores in how many countries?
<Chunks>: {Retrieved document chunks}
<Correct answer>: 12.
<Output>: Crucible is a geodemography computer system created by Tesco, a multinational grocery and general merchandise retailer. Tesco has stores in 12 countries as of recent data, so 12 is the answer.
Example3:
<Question>: What word is in both the genre of Muhammed Suiçmez's band and the genre of Dave Meniketti's band?
<Chunks>: {Retrieved document chunks}
<Correct answer>: Metal.
<Output>: Necrophagist is known for its death metal style. Y&T is often classified under the broader category of heavy metal. So the answer is mental.

{Question}
{Retrieved document chunks}
{Correct answer}
**Output:** {output}

## B Statistics and Experimental Setups for Datasets

Table 4 provides detailed statistics for each dataset, including the number of samples in the training set after curation, test set size, the specific retriever used, and evaluation metrics. Besides, we use the precision@$k$ as an approximation of retrievers' performance. Precision@$k$ is defined as the ratio of chunks that contain the among all retrieved chunks $k$ for each query.

## C Parameter Settings

We detail the key hyperparameters and configurations used across all experiments in Table 5. Specifically, *CML* and **RL** represents contrastive multitask learning and reinforcement learning respectively.

**Table 5:** Parameter settings for experiments. Parameters without being specified are set to their default values as defined by the development package.

| Parameter | Value |
|---|---|
| $\eta$ (CML) | 0.01 |
| $\alpha$ (CML) | 0.5 |
| $\epsilon$ (RL) | 0.2 |
| $\gamma$, $\lambda$(RL) | 0.95 |
| Top-$k$ (RL) | 4 |
| Top-$p$ (RL) | 0.95 |

## D LLM-as-a-Judge For Faithfulness and Completeness Evaluation

### D.1 Prompts for Qwen-2.5-Instruct

To directly evaluate the quality of context generated by ***Oreo*** , we employ Qwen-2.5-Instruct [100] as a reference model to assess two critical dimensions: faithfulness - how well the answer aligns with the retrieved passages, and completeness - to what extend does the generated context cover all essential information to correctly answer the query.

We design instructions for prompting Qwen-2.5-Instruct, as detailed in Table 6 and Table 7.

**Table 6: Qwen-2.5-Instruct used for faithfulness evaluation**

Faithfulness evaluation prompt

**Input:** """You are an expert evaluator assessing the **faithfulness** of a generated context with respect to the original passages. You will be given a query, a set of original passages, and a generated context.
Your task is to determine how accurately the generated context reflects the facts and meanings in the original passages, focusing only on the information required to answering the query. Consider whether the context includes hallucinated information, omits key facts, or misrepresents any content.
Rate the faithfulness on a scale from 0 to 10: - 0: The generated context is entirely unfaithful or unrelated to the original passages. - 10: The generated context is fully faithful, with no hallucinations, distortions, or omissions of relevant information.
Output your score (a float between 0 and 10), followed by a concise explanation of your reasoning. """
**Text:** {text}
**Output:** {Output}

**Table 7: Qwen-2.5-Instruct used for completeness evaluation**

Completeness evaluation prompt

**Input:** """You are an expert evaluator assessing the **completeness** of a generated response. You will be given a query, a set of original passages, and a generated context intended to answer the query.
Your task is to rate how thoroughly the generated context covers all necessary information from the original passages required to answer the query. The context should not omit relevant details, and should avoid adding any external or fabricated content.
Rate the completeness on a scale from 0 to 10: - 0: The generated context provides no useful information for answering the query. - 10: The generated context includes all necessary information to fully and correctly answer the query.
Output your score (a float between 0 and 10), followed by a brief explanation of your reasoning. """
**Text:** {text}
**Output:** {Output}

**Table 4:** Dataset statistics, retrievers and evaluation metrics. EM -Exact Match, F1 - Unigram F1

| Dataset | # Train (k) | # Test (k) | Retriever | Precision@5 | Task | Metric |
|---|---|---|---|---|---|---|
| PopQA | 6.5 | 1.4 | Contriver | 0.287 | Extractive single-hop QA | EM |
| NaturalQuestions | 28.3 | 3.6 | DPR | 0.33 | Extractive single-hop QA | EM |
| TriviaQA | 30.1 | 11.3 | Contriver | 0.43 | Extractive single-hop QA | EM |
| HotpotQA | 20.7 | 5.6 | Contriver | 0.137 | Abstractive multi-hop QA | F1 |
| 2WikiMultiHopQA | 20.7 | 12.6 | BM25 | 0.07 | Abstractive multi-hop QA | F1 |

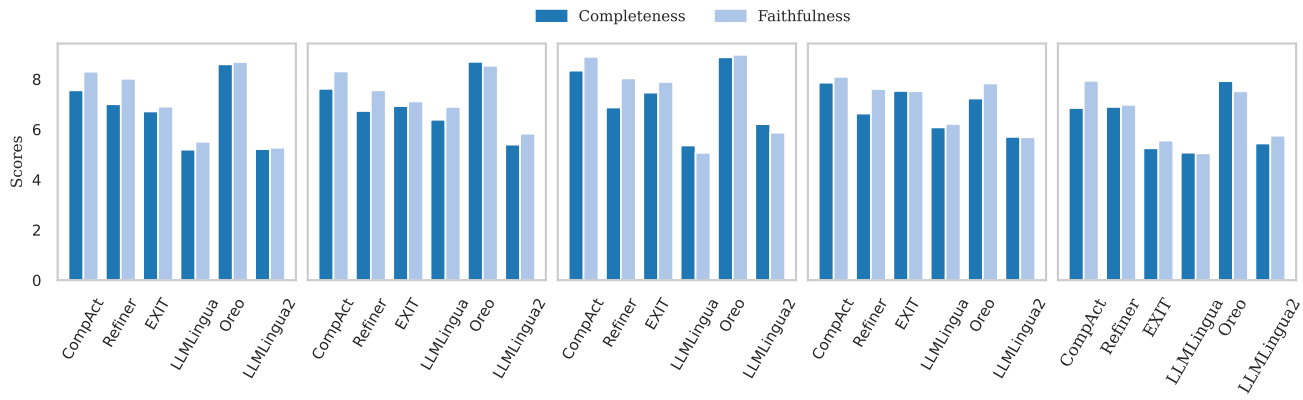


**Figure 7:** Completeness and faithfulness evaluation by Qwen-2.5-Instruct

## D.2 Scoring Results

Figure 7 presents the completeness and faithfulness scores evaluated by Qwen-2.5-Instruct, demonstrating that ***Oreo*** achieves the highest performance on both metrics.

## E Generalizability Evaluation

To evaluate cross-dataset generalizability, we test ***Oreo*** 's transferability by applying models trained on one dataset to a different one without fine-tuning. This assesses ***Oreo*** 's ability to reconstruct and synthesize context under unseen query distributions. Specifically, we evaluate models trained on PopQA for NQ, and on 2WQA for HotpotQA. Detailed results are provided in Table 8.

**Table 8:** QA performance with zero-shot setting. PopQA → NQ represents the model trained on PopQA is applied to NQ.

| Dataset | Model → Dataset | Performance |
|---|---|---|
| NQ | NQ → NQ | 0.4413 |
| | PopQA → PopQA | 0.4682 |
| | PopQA → NQ | 0.4352 |
| HotpotQA | HotpotQA → HotpotQA | 0.6775 |
| | 2WQA → 2WQA | 0.6384 |
| | 2WQA → HotpotQA | 0.6344 |

We perform an ablation study to assess how varying ***Oreo*** 's context length affects downstream performance. By increasing the minimum token threshold from 30 to 300 (in steps of 30) while keeping the top-5 retrieved passages fixed, we observe performance trends summarized in Figure 8.

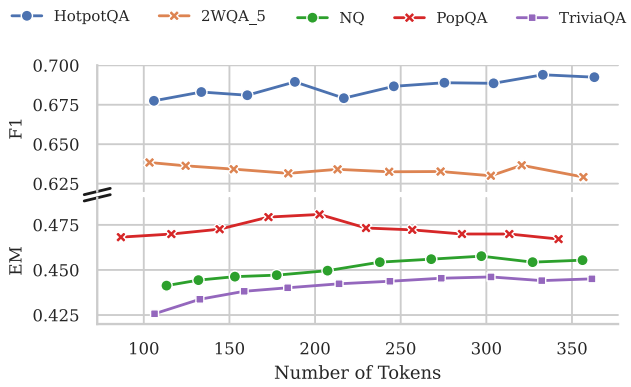


**Figure 8:** Performance of ***Oreo*** generating different lengths of context across five datasets