KO-RAG : Knowledge Organization for Retrieval Augmented Large Language Models with Individual-then-Integrated Feedback

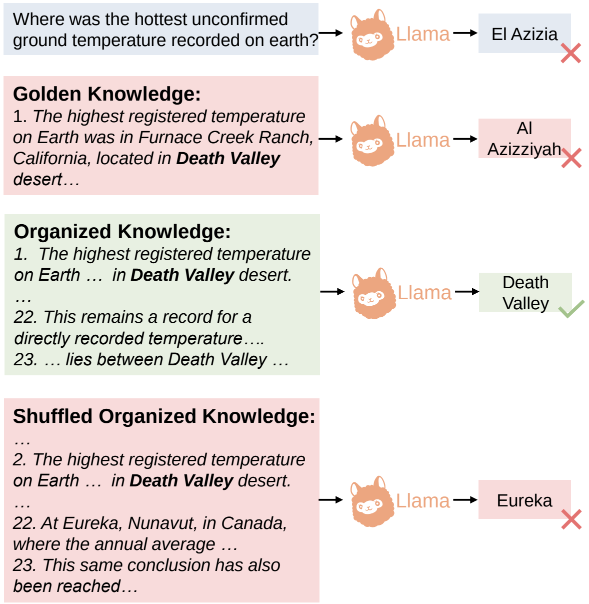
Anonymous ACL submission

Abstract

Retrieval-augmented large language models have shown remarkable potential in knowledge-intensive tasks. However, their performance can be compromised by lengthy, noisy, or irrelevant retrieved information. Recent work focus on knowledge compression, but ignore the feedback from LLM or just incorporate with individual feedback. In this paper, we introduce **KO-RAG**, a knowledge organization method with an external knowledge organization model for retrieval augmented large language models, trained with individual-then-integrated feedback. KO-RAG learns to organize knowledge in a two-stage framework. In the individual feedback stage, our method ranks and filters knowledge by comparing each knowledge, which can measure the helpfulness of each knowledge individually. In the integrated feedback stage, our method organizes the knowledge integratedly by utilizing LLM's feedback on sampled knowledge permutations. Moreover, we design an empty knowledge placeholder to make KO-RAG organize knowledge dynamically. Evaluation on five open-domain question-answering datasets proves that the proposed method has significantly improves the LLMs' performance, outperforming the baseline methods.

1 Introduction

Retrieval-augmented generation (RAG) enhances the large language models (LLMs) ability to provide up-to-date and contextually appropriate responses by incorporating relevant information from curated knowledge bases or the Internet (Nakano et al., 2021; Jiang et al., 2023d; Mallen et al., 2023; Shi et al., 2023). Despite their utility, retrieved texts can be noisy, redundant, and misleading due to imperfect retrievers and incomplete knowledge bases, potentially decreasing performance (Xu et al., 2023; Liu et al., 2024; Cuconasu et al., 2024). Recognizing this shortcoming, researchers have proposed various strategies to compress retrieved knowledge, broadly categorized into two types. The first category is discriminator-based, which relies on a discriminator to judge whether tokens or sentences should be preserved (Jiang et al., 2023c; Li et al., 2023; Xu et al., 2023; Pan et al., 2024). The second method is reranker-based, which uses a scoring model to assign scores to sentences or paragraphs, then selects the top-k highest-scored ones (Glass et al., 2022; Liu, 2022; Huang and Huang, 2024).



Where was the hottest unconfirmed ground temperature recorded on earth?

Golden Knowledge:
*1. The highest registered temperature on Earth was in Furnace Creek Ranch, California, located in **Death Valley** desert…*

Organized Knowledge:
*1. The highest registered temperature on Earth … in **Death Valley** desert.*
…
22. This remains a record for a directly recorded temperature….
23. … lies between Death Valley …

Shuffled Organized Knowledge:
…
*2. The highest registered temperature on Earth … in **Death Valley** desert.*
…
22. At Eureka, Nunavut, in Canada, where the annual average …
23. This same conclusion has also been reached…

Figure 1: An example to show the necessary of knowledge organization. In the example, Llama's responses are different to the same question under different knowledge contexts. **Golden Knowledge**: Selected both by human annotators and Llama feedback, which also contains the right answer of the given question, yet fails to elicit the correct answer. **Organized Knowledge**: Includes information not directly related to the question, but enables Llama to answer correctly. **Shuffled Organized Knowledge**: Randomly reordered version of the organized knowledge, leading to an incorrect prediction.

Realizing the gap between pre-defined or heuristic based and LLM's preference, recent research has explored using LLM's feedback to train discriminators or rerankers (Xu et al., 2023; Jiang et al., 2023b). However, these approaches typically operate at a individual level, which use LLM's feedback to measure one retrieved document's helpfulness, but not take all the knowledge context into consideration. While such method is straightforward to implement, but often leads to suboptimal results. As illustrated in Figure 1, the golden knowledge—both chosen by human annotators and the LLM itself[1]—may not always enable the LLM to generate the correct answer.[2] Our research demonstrates that organized knowledge, which includes information about "Death Valley" but is not directly helpful to answer the question, can actually help the LLM answer correctly. Moreover, we observed that shuffling this organized knowledge can mislead the LLM[3]. These observations underscore the critical role of LLM's integrated feedback in knowledge content selection and arrangement.

Building on these insights, we introduce the concept of ***knowledge organization***. We define this as the process of searching the optimal ordered subsequence of original knowledge that maximizes an LLM's probability of answering correctly. Given that the precise mechanism by which LLMs integrate retrieved contextual knowledge with their parametric knowledge remains unclear, and that searching the optimal among all possible candidates is an NP-Hard problem, we propose **KO-RAG**, an external knowledge organization model with a two-stage training framework to investigate this problem. In the first stage, we rank knowledge based on individual LLM feedback, selecting those that demonstrably improve the LLM's ability to answer correctly. The second stage involves calculating the LLM's knowledge context preference using integrated feedback and optimizing the model through Direct Preference Optimization (DPO, Rafailov et al., 2024). To enable dynamic knowledge ranking, we introduce the concept of an empty knowledge placeholder, allowing for flexible sequence lengths and more nuanced organization strategies.

We conducted experiments across five knowledge-intensive tasks, including open-domain question answering and long-form question answering. Our experimental results demonstrate that, given similar input knowledge length constraints, our method outperforms both discriminator-based and reranker-based baseline approaches. These findings suggest that our knowledge organization technique more effectively leverages available information, leading to improved performance on language understanding and generation tasks.

We conclude our contribution as follows:

- We propose KO-RAG, a two stage framework that enables model to organize knowledge via LLM's individual and integrated feedback.
- We model knowledge organization in a unified process with proper distribution estimation by introducing empty knowledge placeholder.
- Experiments on various knowledge intensive tasks proved the efficiency of our proposed method.

2 Problem Formulation

In the basic RAG framework, we have a question Q, its corresponding answer A, and knowledge retrieved from a knowledge base $K = \{k_1, k_2, \cdots, k_n\}$, where k_i denotes a knowledge sentence. We consider the knowledge organization process as a picking and ranking task, which can be viewed as finding a specific permutation of knowledge. Let $O = \{O_1, O_2, \cdots, O_m\}, m = \sum_{i=0}^{n} \frac{n!}{(n-i)!} = \lfloor e * n! \rfloor$ represent all possible permutations of retrieved knowledge, where O_i denotes one partial permutation of $\{1, 2, \cdots, n\}$, e denotes the Euler number and $\lfloor . \rfloor$ denotes the floor function. The target $O_{optimal}$ of our knowledge organization process can be defined as follows:

$$O_{optimal} := \max_{O_i \in O} P(A|Q, K_{O_i}) \qquad (1)$$

where K_{O_i} denotes the knowledge under the order O_i. This formulation seeks to find the permutation of knowledge that maximizes the probability of generating the correct answer A given the question Q and the organized knowledge K_{O_i}.

3 Method

The vast search space, with $O(n!)$ possible candidates, renders exhaustive search computationally infeasible. To navigate these complexities, we employ a sophisticated, two-stage training methodol-

[1] Given the knowledge and question, calculate the conditional probability that the LLM predict the correct answer.

[2] This phenomenon is known as context-parameter knowledge conflict (Tan et al., 2024; Xie et al., 2024) or extrinsic hallucinations (Huang et al., 2023; Ji et al., 2023).

[3] This finding is also corroborated by recent studies (Cuconasu et al., 2024; Liu et al., 2024)

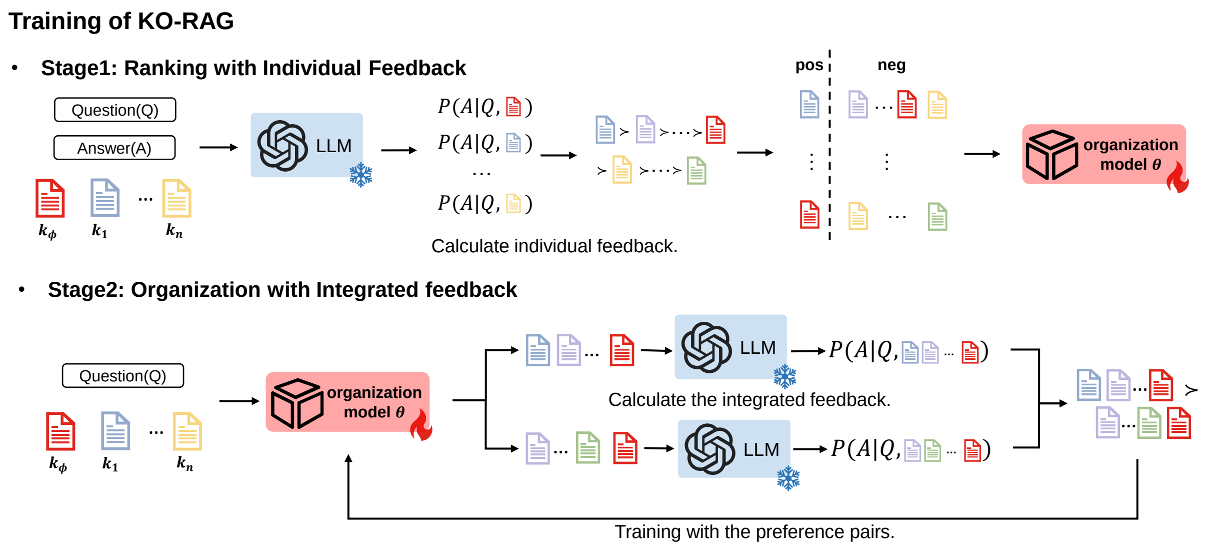

Training of KO-RAG
Stage1: Ranking with Individual Feedback
Question(Q)
Answer(A)
k_ϕ
k_1
...
k_n
LLM
$P(A|Q,$ $)$
$P(A|Q,$ $)$
...
$P(A|Q,$ $)$
Calculate individual feedback.
pos
neg
organization model θ
Stage2: Organization with Integrated feedback
Question(Q)
k_ϕ
k_1
...
k_n
organization model θ
LLM
$P(A|Q,$ $)$
Calculate the integrated feedback.
LLM
$P(A|Q,$ $)$
Training with the preference pairs.

Figure 2: Overview of training process of the KO-RAG framework. k_i denotes the retrieved knowledge and the empty knowledge placeholder is denoted by K_ϕ. In the stage 1, LLM provides feedback to rank each retrieved knowledge and the empty knowledge placeholder. In the stage 2, KO-RAG sample 2 different knowledge rank lists based on the scores predicted by KO-RAG, in which the empty knowledge placeholder serves as the end of list. KO-RAG is optimized with the integrated feedback of two knowledge lists and DPO.

ogy of knowledge organization. The overall procedure is illustrated in Figure 2, which includes Ranking with Individual Feedback (§3.1) and Organization with Integrated Feedback (§3.2).

3.1 Ranking with Individual Feedback

Training procedure Following the definition in Section 2, we can seek an order $O_{id} = \{o_1, o_2, \cdots, o_n\}$, such that $\forall i > j$ $P(A|Q, k_{o_i}) > P(A|Q, k_{o_j})$, which relies on LLM's individual feedback. To remove unhelpful knowledge, we introduce an empty knowledge placeholder ϕ, where $P(A|Q, \phi) = P(A|Q)$. This allows comparison of each knowledge sentence's utility against a baseline of no additional knowledge. Hence we obtain a knowledge rank based on the individual feedback $O_{id} = \{o_1, o_2, \cdots, o_{n+1}\}$ and $k_{o_m} = k_\phi$. Then we consider all knowledge rank higher than the empty knowledge placeholder (including the empty knowledge placeholder) as positive examples. For each positive knowledge sentence K_{o_i}, we sample T lower-ranked knowledge sentences as negative samples and construct the training dataset D with Q and O_{id} as follows.

$$D = \{(Q, o_p, o_{n_1,\cdots,n_T})|o_p, o_{n_i} \in Q_{id} \& o_p > o_{n_i}\} \quad (2)$$

based on the corresponding query Q and the knowledge k_{o_i}, the score model $\mathcal{F}$ will predict the score $s_{o_i} := \mathcal{F}(Q, k_{o_i})$. The training loss which is InfoNCE loss (Oord et al., 2018), is defined as follows:

$$L = \sum_{(Q,o_p,o_{n_1,\cdots,n_T})\in D} \frac{exp(s_{o_p}/\tau)}{exp(s_{o_p}/\tau) + \sum_{j=1}^{T} exp(s_{o_{n_j}}/\tau)} \quad (3)$$

Where τ is a temperature parameter controlling the sharpness of the probability distribution.

Score Model Following recent approaches in the field (Chen et al., 2024; Pan et al., 2024), we employ an encoder architecture as our score model. Given an input query Q and a knowledge sentence k_i, the model calculates a helpfulness score s_i as follows:

$$s_i = Enc(Emb(Q \circ k_i)) \quad (4)$$

where Enc represents the encoder function, $Emb()$ denotes the embedding output and $\circ$ denotes string concatenation.

Virtual Tokens for Empty Knowledge Placeholder The empty sequence presents a significant challenge in the training process due to its format contrasts with normal "question-knowledge" pairs, as it lacks any knowledge text. To address this issue, we draw inspiration from Prefix-Tuning techniques (Li and Liang, 2021), and introduce additional virtual tokens to represent the empty knowledge placeholder. These virtual tokens serve as a proxy for the absent content, providing a more consistent input structure across all cases. The calculation process of empty knowledge placeholder K_ϕ is defined as follows:

$$s_\phi = Enc(V \oplus Emb(Q)) \tag{5}$$

where $V := [V_0, V_1, \cdots, V_{m-1}] \in \mathbb{R}^{m\times d}$ represents m randomly initialized virtual tokens, each with dimension d, $\oplus$ denotes vector concatenation and $Emb()$ denotes the embedding output.

3.2 Organization with Integrated Feedback

Training procedure We begin by defining a preference relationship between knowledge orders. Given a question Q, its corresponding answer A, and two predicted knowledge orders O_i and O_j, we define the preference as follows

$$O_i \succ O_j \ if \ P(A|Q, K_{O_i}) > P(A|Q, K_{O_j}) \tag{6}$$

in which K_{O_i} denotes the knowledge under the order O_i.

Based on this preference relationship, we can construct a training dataset $D := (Q, O_w, O_l)|O_w, O_l \in O, O_w \succ O_l$, where O_w is the preferred (winning) order and O_l is the less preferred (losing) order for a given question. Rather than pre-building the entire training dataset, we adopt an online reinforcement learning paradigm.

We employ DPO to train our model using feedback from LLM's feedback. Let P_{ref} and P_θ denote the probability distributions estimated by the reference model and the model being trained, respectively. Given a preferred order O_w and a less preferred order O_l, the DPO loss function is defined as:

$$L_{DPO} = -\log\sigma(\beta\log\frac{P_\theta(O_w)\cdot P_{ref}(O_l)}{P_{ref}(O_w)\cdot P_\theta(O_l)}) \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function and β is a constrained hyper-parameter.

To enhance training stability and maintain high confidence in preferred orders, we introduce an additional Maximum Likelihood Estimation (MLE) loss for the preferred order O_w:

$$L_{MLE} = -\log P_\theta(O_w) \tag{8}$$

this MLE loss encourages the model to assign high probabilities to preferred orders. We combine the DPO and MLE losses to form our final loss function:

$$L = L_{DPO} + \lambda L_{MLE} \tag{9}$$

where λ is a hyper-parameter that balances the contribution of the MLE loss relative to the DPO loss.

Distribution estimation of $P(O)$ We first introduce the distribution estimation of the permutation. Given a query q and a set of retrieved knowledge $K = \{k_1, k_2, \cdots, k_n\}$, the score model M will calculate the scores of each text $S = \{s_i|s_i = M(q, k_i)\}$. According to the Plackett-Luce model, we can define the probability of a full ranking order $O_f = \{o_{f_1}, o_{f_2}, \cdots, o_{f_n}\}$ as follows:

$$\begin{aligned} P(O_f|x) &= \prod_{i=1}^{n} P(o_{f_i}|o_{<f_i}, x) \\ &= \prod_{i=1}^{n} \frac{exp(s_{f_i})}{\sum_{j=i}^{n} exp(s_{f_j})} \end{aligned} \tag{10}$$

extending this concept, we propose a method to calculate the probability of a partial ranking order $O_p = \{o_{p_1}, o_{p2}. \cdots, o_{p_m}\}$, where $m \le n$ and $k_{o_{p_m}} = \phi$:

$$\begin{aligned} P(O_p|x) &= \prod_{i=1}^{m} P(o_{p_i}|o_{<p_i}, x) \\ &= \prod_{i=1}^{m} \frac{exp(s_{p_i})}{\sum_{p_j \notin \{o_{<p_i}\}} exp(s_{p_j})} \end{aligned} \tag{11}$$

for a detailed proof, please refer to Appendix A. This formulation allows us to treat the passage scoring process as a sequential decision-making problem, opening the possibility of optimizing it using reinforcement learning techniques.

3.3 Inference Stage

During inference, we employ a greedy method to select and order the knowledge. Given a query q and a set of retrieved knowledge passages $K = \{k_1, k_2, \cdots, k_n\}$, our scoring model M calculates a score for each passage, producing a set of scores $S = \{s_i|s_i = M(q, k_i)\}$. We then construct the predicted order $O = \{o_1, o_2, \cdots, o_m\}$ such that: $\forall i > j \ s_{o_i} > s_{o_j}$ and $k_{o_m} = \phi$ (the empty knowledge placeholder).

4 Experiment setup

Datasets To evaluate the effectiveness of our proposed method, we conduct experiments on several knowledge intensive datasets. Specifically, we conduct experiments on three open domain question answering datasets: Natural Questions (NQ, Kwiatkowski et al. 2019), AmbigNQ (Min et al., 2020), PopQA[4] (Mallen et al., 2023) and two long-

[4] Since there is no official train-test split of PopQA, we use top-13000 as the train set and 1267 as test set.

Dataset	Train	Test	Avg token
NQ	87372	2837	3168
AmbigNQ	19363	5749	3177
PopQA	13000	1267	3237
ASQA	4353	948	3169
ELI5	272633	1507	3021

Table 1: Statistical information of datasets. We report the number of question-answer pairs in train dataset and test dataset in "Train" and "Test". We also introduce the average number of tokens of retrieved knowledge in "Avg token", which includes train set and test set.

form question answering datasets: ASQA (Stelmakh et al., 2022), ELI5 (Fan et al., 2019). The statistical information is listed in Table 1

Baselines In this paper, we consider the following three categories baselines:

- Naive baselines: **Zero-shot**, without any context knowledge. **RAG**, we use top-50% knowledge scored by the retriever.
- Discriminator-based baselines for RAG: **Selective-Context (SC)** (Li et al., 2023), **LongLLMLingua (LLingua)** (Jiang et al., 2023b), **LLMLingua2(Lingua-2)** (Pan et al., 2024). For discriminator-based baselines, we control the compression rate to 50%, which means deleting 50% of retrieved knowledge in token-level.
- Reranker-based baselines for RAG: **BM25** (Robertson et al., 1995), **RankT5 (T5)** (Zhuang et al., 2023), **BGE-raranker (BGE)** (Chen et al., 2024). For reranker-based baselines, we use reranker to do a sentence-level rerank and pick top 50%.

Implementation Detail In experiment, we use Llama2-13B-chat (Touvron et al., 2023) as the base model to provide feedback and as a answer generator in RAG framework. We also test our method based on Llama-2-7B-chat (Touvron et al., 2023) and Mistral-7B-Instruct (Jiang et al., 2023a). For knowledge retrieval, we use contriever (Izacard et al., 2021) as a knowledge retriever, and the 2019/08/01 Wikipedia dump pre-processed by Petroni et al. (2021) as a knowledge base. For each question, we retrieve top-20 chunks as retrieved knowledge. The model architecture is XLM-roberta-large (Conneau et al., 2020). Hyperparameter and more details are listed in Appendix B. Prompts we use are listed in Appendix D.

Metric For open-domain question answering, we use fuzzy **Exact Match(EM)** to measure whether generated answers contains the golden answers. And for long-form question answering, we use precision-based metric **BLEU-1(B-1**, Papineni et al. 2002) to calculate the similarity between generated answers and golden answers. Besides reporting the performance, we also report **average knowledge token(#token)**[5] of each method.

5 Results and Discussion

5.1 Main result

The experimental results are presented in Table 2. Our proposed method outperforms all baseline methods across nearly all datasets with the shortest average context length, with the exception of ELI5. In ELI5, our proposed method achieve the best performance with a comparable average context length with discriminator-based and reranker-based knowledge compression method.

Comparison with Discriminator-based Methods Selective-context and LLMLingua2, designed for universal context compression rather than specifically for RAG, demonstrate relatively lower performance. LongLLMLingua, which is tailored for RAG and utilizes Llama2-7B for knowledge compression, exhibits strong performance but still falls short of our proposed method.

Comparison with Reranker-based Method Model-based reranker methods show impressive performance on open-domain question answering tasks but appear less effective for long-form question answering. N-gram based methods like BM25 perform poorly on open-domain question answering and long-form question answering scenarios. In terms of overall performance across diverse tasks, our proposed method consistently outperforms these approaches.

5.2 The Effectiveness of Integrated Feedback

To assess the impact of integrated feedback from LLMs, we conducted a comparative analysis of our system's performance before and after the integrated feedback stage. The results are presented in Table 3. Our findings demonstrate that the incorporation of integrated feedback leads to further performance improvements across all the datasets.

[5] In this paper, all the token refers to the text tokenized by Llama2 tokenizer.

	NQ		AmbigNQ		PopQA		ASQA		ELI5	
	EM	#token	EM	#token	EM	#token	B-1	#token	B-1	#token
Naive										
Zero-shot	43.32	0	32.75	0	11.13	0	11.39	0	13.93	0
RAG	45.51	1576	36.49	1588	36.54	1613	10.40	1585	14.22	**1511**
Discriminator-based baselines for RAG										
SC (Li et al., 2023)	43.57	1890	33.55	1897	34.65	1987	10.33	1893	14.25	1795
LLingua (Jiang et al., 2023b)	47.83	1714	39.73	1729	39.07	1758	10.82	1722	14.22	1659
Lingua2(Pan et al., 2024)	46.00	1788	35.43	1808	35.44	1829	10.98	1803	14.14	1696
Reranker-based baselines for RAG										
BM25 (Robertson et al., 1995)	43.56	1838	36.26	1861	34.65	1823	9.83	1841	13.88	1769
T5 (Zhuang et al., 2023)	46.67	1757	37.85	1725	37.17	1716	10.42	1756	14.03	1578
BGE (Chen et al., 2024)	47.76	1787	37.82	1837	36.54	1863	10.38	1841	14.22	1692
KO-RAG	**50.58**	**1350**	**42.29**	**1366**	**42.46**	**614**	**11.96**	**1206**	**14.74**	1773

Table 2: Performance comparison on knowledge-intensive tasks. This table presents the evaluation results across various knowledge-intensive tasks. The **bold** values indicate the best performance for each metric, while underlined values represent the second-best performance. For context length, **bold** figures denote the shortest number of tokens, and underlined figures indicate the second shortest. Note that the Zero-shot method is excluded from this token count comparison due to its unique nature

Dataset	w/o. IF	w. IF	baseline
NQ	47.87	50.58	47.83
AmbigNQ	38.20	42.29	39.73
PopQA	40.81	42.46	39.07
ASQA	10.25	11.96	11.39
ELI5	13.76	14.74	14.25

Table 3: Impact of integrated feedback(IF) on model performance. "w/o. IF“ denotes the performance without integrated feedback and "w. IF" denotes the performance after organization with integrated feedback. "baseline" denotes the best performance of baseline method.

Notably, even in the absence of the integrated feedback stage, our proposed method outperforms the baseline approach on several datasets, including NQ and PopQA. This underscores the effectiveness of our initial knowledge reorganization stage. The dual benefits observed—improvements from both the initial knowledge reorganization and the subsequent integrated feedback stage—suggest that our two-stage approach offers a powerful framework for enhancing LLM performance.

5.3 Fine-grained Performance Analysis

While knowledge organization and compression can lead to overall performance improvements, they may also cause changes at the individual sample level. To analyze this, we categorize questions into four groups based on their pre- and post-reorganization performance: PP (correct before and after), NP (incorrect before, correct after), PN (correct before, incorrect after), and NN (incorrect before and after). We then introduce a fine-grained efficiency score S, defined as:

$$S = \frac{|NP|}{|PN|} \tag{12}$$

where $|NP|$ and $|PN|$ denotes the sample number of NP and PN group. A higher S value indicates that when the model potentially distorts one sample (PN), it can rectify a larger number of samples (NP), thus demonstrating higher efficiency. We calculate this efficiency score across three datasets: NQ, AmbigNQ, PopQA. Our experimental results, which are listed in table 4 that demonstrate our proposed method consistently maintains higher efficiency scores across all datasets compared to baseline approaches. This suggests that our knowledge reorganization technique not only enhances overall accuracy but also achieves this improvement more efficiently, with a better trade-off between correcting previously incorrect answers and maintaining correct ones.

5.4 Generalize to Different LLMs

Our method is trained using feedback from a specific LLM. A natural question arises: can this method generalize to different models? To address this, we evaluated our approach on three open-domain question answering datasets using two distinct models: Llama2-7B (Touvron et al.,

	NQ	AmbigNQ	PopQA	Avg
SC	0.81	0.69	0.79	0.74
LLingua	<u>1.30</u>	<u>1.57</u>	<u>1.51</u>	<u>1.47</u>
Lingua2	1.05	0.87	0.85	0.93
BM25	0.82	0.97	0.75	0.89
T5	1.14	1.20	1.10	1.17
BGE	1.27	1.19	1.00	1.20
KO-RAG	**1.65**	**2.02**	**2.07**	**1.89**

Table 4: The efficiency score of different method across different datasets. We **bold** the highest stable score and <u>underline</u> the score highest score.

2023) and Mistral-7B (Jiang et al., 2023a). We present a summary of our findings in Table 6 for Llama2-7B-chat and Table 5 for Mistral-7B[6]. The results demonstrate that our method consistently outperforms all baseline approaches across both models, indicating strong generalization capabilities. Interestingly, we observed that the performance gap between our method and the baselines narrows when applied to Mistral-7B, compared to its performance with Llama2 series models. This suggests that the different LLMs may have similar knowledge preference, thus enabling our method generalize to these LLMs.

	NQ	AmbigNQ	PopQA
Zero-shot	46.21	37.35	31.41
RAG	50.26	39.73	40.41
SC	47.30	35.76	36.70
LLingua	<u>52.73</u>	<u>43.19</u>	<u>43.09</u>
Lingua2	51.00	38.28	40.02
BM25	48.82	38.41	36.54
T5	51.64	40.27	40.25
BGE	<u>52.73</u>	40.86	40.57
KO-RAG	**54.04**	**43.50**	**45.54**

Table 5: Performance on open-domain question answering tasks, which is based on Mistral-7b. The **bold** and <u>underlined</u> values indicate the best and the second-best performance for each metric, respectively.

5.5 Model Context Preference Estimation

In this section,we address two critical questions: "Is our trained model an effective estimator of LLM's context preferences?" and "Does reinforcement learning improve the model's ability to estimate LLM preferences?". To investigate these issues, we compare the agreement and Cohen's kappa (Cohen, 1960) between KO-RAG and LLMs. Specifically, we reuse the LLMs' preference defined in equation 6 and KO-RAG's preference defined in equation 11. Figure 3 illustrates our findings. Interestingly, while reinforcement learning does not significantly improve the raw agreement rate, it does enhance the Cohen's kappa score. This suggests that the reinforcement learning process refines the model's ability to capture more nuanced aspects of LLM preferences, beyond simple binary agreement. However, it is important to note that both the agreement rate and Cohen's kappa remain below satisfactory levels. This indicates that while our approach shows promise, there is still considerable room for improvement in accurately modeling LLM context preferences.

	NQ	AmbigNQ	PopQA
Zero-shot	41.28	30.27	28.97
RAG	43.11	34.27	36.23
SC	40.01	31.41	30.54
LLingua	<u>44.70</u>	<u>36.88</u>	<u>37.02</u>
Lingua2	43.29	33.74	32.99
BM25	40.82	32.63	32.83
T5	42.47	34.61	35.44
BGE	43.18	34.04	35.28
KO-RAG	**47.44**	**40.22**	**43.49**

Table 6: Performance on open-domain question answering tasks, which is based on Llama2-7b. The **bold** and <u>underlined</u> values indicate the best and the second-best performance for each metric, respectively.

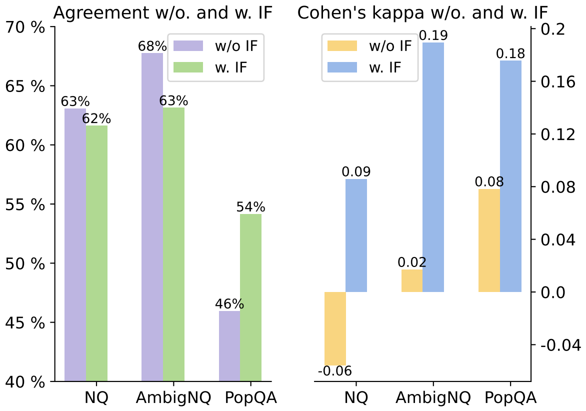


Figure 3: Agreement rate and Cohen's kappa between KO-RAG and LLM preferences w/o. and w. integrated feedback (IF) across NQ, AmbigNQ, and PopQA datasets

5.6 Case Study

Table 7 illustrates examples comparing model performance using contexts processed by KO-RAG versus silver knowledge contexts (The optimization target in Stage 1, refer to Chapter 3.1). Consider

[6] We use the Mistral-7B-instruct-v0.2 version

the question, "*Who plays the science officer on Star Trek: Discovery who is also a chief engineer?*" When the initially retrieved knowledge lacks the correct answer, the silver context fails to guide the LLM towards an accurate response. In contrast, KO-RAG demonstrates its effectiveness by successfully leveraging the LLM's inherent parameter knowledge that "*Paul Stamets is played by actor Anthony Rapp.*" By effectively filtering and reorganizing this information, KO-RAG enables the LLM to provide the correct answer: Anthony Rapp.

Question:
Who plays the science officer on Star Trek discovery who is a chief engineer?

Standard answer:
"Anthony Rapp", "Anthony Deane Rapp"

Silver context:
1: Yelchin died in a car accident on June 19, 2016,
2: ... specifically from Captain Christopher Pike, ...

LLM's prediction:
The character who plays the science officer and chief engineer on Star Trek: Discovery is Captain Christopher Pike, portrayed by Anson Mount.

KO-RAG context:
1: Yelchin died in a car accident on June 19, 2016,
2. Science Officers Saru and Stamets, respectively.
3. After the first season concluded with the ...
...
6. ... the relationship between Culber and Stamets would continue to be explored.

LLM's prediction:
The character who plays the science officer and chief engineer on Star Trek: Discovery is Paul Stamets, played by actor Anthony Rapp.

Table 7: Case comparison between silver context and KO-RAG processed context On AmbigNQ dataset. Entities that appear in both the context and the LLM's prediction are highlighted in red.

6 Related Work

Retrieval Augmentation Generation Retrieval augmented generation, which uses retrieved knowledge as generation context, significantly improves the the accuracy, credibility and interpretability of generated texts (Gao et al., 2023; Ren et al., 2023; Wang et al., 2023b; Louis et al., 2024). One mainstream method is training retriever and LLMs end-to-end (Guu et al., 2020; Wang et al., 2024a; Nakano et al., 2021; Asai et al., 2023; Borgeaud et al., 2022; Wang et al., 2023a). Another method freezes the LLMs and uses retrieved knowledge as additional context with various strategies including retrieval query refinement, structured knowledge indexing, and iterative retrieval mechanisms (Ma et al., 2023; Jiang et al., 2023d; Wang et al., 2024b). Nonetheless, the inclusion of retrieved data can introduce new challenges, as it often yields noisy or redundant information that might distract the LLMs from pertinent content (Yoran et al., 2023; Liu et al., 2024).

Knowledge Compression For RAG Knowledge compression could remove the irrelevant contexts and reduce the input context length, thus improving the model's performance and decrease the cost of inference. One effective method is reranker-based, which use a reranker model to modeling the relevance between retrieved knowledge and question, then remove the less relevant knowledge to improve performance and reduce context length (Glass et al., 2022; Izacard et al., 2023; Wang et al., 2023c; Xu et al., 2023; Huang and Huang, 2024). Another main method is discriminator based, which relies on a discriminator to determine which part should be deleted, including sentence-level self-information calculated by a small language model (Li et al., 2023; Jiang et al., 2023c), token-level discriminator (Pan et al., 2024) or combine token-level discriminator and reranker (Jiang et al., 2023b). But these method ignores LLM's feedback or just utilize LLM's feedback in a individual level with greedy search. Recently, there are several studies to tackle this problem with a Seq2Seq model (Yang et al., 2023; Jin et al., 2024; Zhu et al., 2024). While these methods are easily trained with LLM's feedback, they have different drawbacks inherited from Seq2Seq architecture, such as generation hallucinations, limited input context and could not inference in parallel. Since these method have different drawbacks and model architectures, we do not consider them as baselines.

7 Conclusion

We introduce KO-RAG, an advanced knowledge organization model to enhance RAG system. Our method is accomplished with two-stage training framework, utilizing LLM's individual and integrated feedback respectively. Our comprehensive experiments across diverse open-domain and long-form question answering datasets demonstrate the efficacy of our method. Through in-depth analysis, we elucidate the benefits of the integrated feedback, and highlight our method's fine-grained efficiency and its ability to generalize across various LLMs.

Limitation

We conclude the limitation of our method as follows: First, we test our method in various open domain question answering datasets and long form question answering datasets, but the questions in these datasets are not complex. For complex questions, which need LLM to perform multi-step reasoning and iterative retrieval, whether our method still works remains to be examined. Secondly, our method relies on training with LLM's feedback, thus making it more time costing and GPU costing than the heuristic method such as Selective-Context (Li et al., 2023) and LLMLingua (Jiang et al., 2023c).

Acknowledgements

References

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

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*, pages 2206–2240. PMLR.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

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. *arXiv preprint arXiv:2401.14887*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

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*.

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*, pages 2701–2715.

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

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Yizheng Huang and Jimmy Huang. 2024. A survey on retrieval-augmented text generation for large language models. *arXiv preprint arXiv:2404.10981*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

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

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

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

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023d. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Jiajie Jin, Yutao Zhu, Yujia Zhou, and Zhicheng Dou. 2024. Bider: Bridging knowledge inconsistency for efficient retrieval-augmented llms via key supporting evidence. *arXiv preprint arXiv:2402.12174*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

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

Jerry Liu. 2022. LlamaIndex.

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

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22266–22275.

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*, pages 5303–5315.

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)*, pages 9802–9822.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

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.

Zixuan Ren, Yang Zhao, and Chengqing Zong. 2023. Towards informative open-ended text generation with dynamic knowledge triples. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3189–3203.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288.

Hexiang Tan, Fei Sun, Wanli Yang, Yuanzhuo Wang, Qi Cao, and Xueqi Cheng. 2024. Blinded by generated contexts: How language models merge generated and retrieved contexts for open-domain qa? *arXiv preprint arXiv:2401.11911*.

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

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, et al. 2023a. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7763–7786.

Junda Wang, Zhichao Yang, Zonghai Yao, and Hong Yu. 2024a. Jmlr: Joint medical llm and retrieval training for enhancing reasoning and professional question answering capability. *arXiv preprint arXiv:2402.17887*.

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

Yubo Wang, Xueguang Ma, and Wenhu Chen. 2023b. Augmenting black-box llms with medical textbooks for clinical question answering. *arXiv preprint arXiv:2309.02233*.

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

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. 2023. Recomp: Improving retrieval-augmented lms with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*.

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*, pages 5364–5375.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Kun Zhu, Xiaocheng Feng, Xiyuan Du, Yuxuan Gu, Weijiang Yu, Haotian Wang, Qianglong Chen, Zheng Chu, Jingchang Chen, and Bing Qin. 2024. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. *arXiv preprint arXiv:2406.01549*.

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*, pages 2308–2313.

A Detail Proof of the Extend Plackett-Luce Model

Given retrieved knowledge $K = \{k_1, k_2, \cdots, k_n\}$ and responding scores $S = \{s_1, s_2, \cdots, s_n\}$, according to Plackett-Luce model, the probability distribution of a full rank $O_i = \{o_1, o_2, \cdots, o_n\}$ defined is as follows:

$$
\begin{aligned}
P(O_i) &= \prod_{i=1}^{n} P(o_i|o_{<i}) \\
&= \prod_{i=1}^{n} \frac{exp(s_{o_i})}{\sum_{j=i}^{n} exp(s_{o_j})}
\end{aligned} \tag{13}
$$

Then, we consider the empty case $\phi \in K$, the predicted extractive rank $E_i = \{e_0, e_1, \cdots, e_m\}$ that satisfying $m \leq n$ and $k_{e_m} = \phi$. In this case, we consider the remain index set $R = \{r_i | i = 1, 2, \cdots, n \; and \; r_i \notin E_i\}$ and remain score set $S_R = \{s_i | i \in R\}$. Then we have

Then, considering the empty knowledge as a special case, we have a knowledge $K = \{k_0, k_1, k_2, \cdots, k_n\}, k_0 = \phi$ and its responding scores $S = \{s_0, s_1, s_2, \cdots, s_n\}$. For a partial rank $E_i = \{e_0, e_1, \cdots, e_m\}$ that satisfying $m \leq n + 1$ and $k_{e_m} = \phi$, its probability distribution is

$$P(E_i) = \prod_{i=0}^{m} \frac{exp(s_{e_i})}{\sum_{j \notin \{o_{<e_i}\}} exp(s_j)} \tag{14}$$

To approve this, we consider the remaining knowledge set $R = \{k_i | i = 1, 2, \cdots, n \; and \; k_i \notin E_i\}$, the remaining scores set $S_R = \{s_i | k_i \in R\}$ and R's all possible permutation $O_R = \{O_{R_i}\}$. The partial rank E_i and one permutation of $O_{R_i} \in O_R$ make up a full rank of $O_i \in O$. Noting that two different permutation of K, which share the same partial rank E_i, shows no difference to LLM, the probability of E_i should be the sum of probability of permutation $O_i \in O$ which starts with E_i. In other words, we have :

$$\begin{aligned} P(E_i) &= \sum_{O_{R_i} \in O_R} P(E_i \bigoplus O_{R_i}) \\ &= \sum_{O_{R_i} \in O_R} \prod_{i=0}^{m} \frac{exp(s_{e_i})}{\sum_{j=i}^{m} exp(s_{e_j}) + \sum_{j=1}^{n-m+1} exp(s_{r_j})} \\ &\prod_{j=1}^{n-m+1} \frac{exp(s_{r_i})}{\sum_{j=i}^{n-m+1} exp(s_{r_j})} \\ &= \prod_{i=0}^{m} \frac{exp(s_{e_i})}{\sum_{j=i}^{m} exp(s_{e_j}) + \sum_{j=1}^{n-m+1} exp(s_{r_j})} \\ &\sum_{O_{R_i} \in O_R} \prod_{j=1}^{n-m+1} \frac{exp(s_{r_i})}{\sum_{j=i}^{n-m+1} exp(s_{r_j})} \end{aligned} \tag{15}$$

in which $\bigoplus$ means concatenation of two sequence.

We notice that $\frac{exp(s_{r_i})}{\sum_{j=i}^{n-m+1} exp(s_{r_j})}$ is the probability of O_{R_i} under the set R, which simplify the equation as

$$\begin{aligned} P(E_i) &= \prod_{i=0}^{m} \frac{exp(s_{e_i})}{\sum_{j=i}^{m} exp(s_{e_j}) + \sum_{j=1}^{n-m+1} exp(s_{r_j})} \\ &\sum_{O_{R_i} \in O_R} P(O_{R_i}) \\ &= \prod_{i=0}^{m} \frac{exp(s_{e_i})}{\sum_{j=i}^{m} exp(s_{e_j}) + \sum_{j=1}^{n-m+1} exp(s_{r_j})} \\ &= \prod_{i=0}^{m} \frac{exp(s_{e_i})}{\sum_{j \notin \{o_{<e_i}\}} exp(s_j)} \end{aligned} \tag{16}$$

B Implementation Details

B.1 Datasets

Natural Question (NQ) is a corpus of real questions issued to the Google search engine. We use the data processed by KILT.

AmbigNQ collects questions from NQ and rewrite the questions to solve the ambiguity. We use the official released data.

PopQA collects knowledge triples from Wikidata and convert the knowledge triple to a question-answer pair with templates. We use the official released data and use the top-13000 as the train set, remaining 1267 as the test set.

ASQA is a dataset of high-quality long-form answers to 6,316 ambiguous factoid questions. We use the official released data.

ELI5 collects question-answer pair from a subreddit from Reddit, named as Explain Like I'm Five(ELI5). We use the data processed by KILT.

B.2 Hyper parameter

In stage 1, we search the learning rate in {2e-4, 1e-4, 5e-5, 1e-5}, and the learning rate for NQ is 2e-4, AmbigNQ is 1e-4, PopQA is 5e-5, ASQA is 1e-5 and eli5 is 1e-5. We set the number of batch size as 8 and the number of negative samples as 31. we set the gradient accumulation steps as 2. We set the number of virtual token in equation 5 as 50.

In stage 2, we set the learning rate as 1e-6, the β is 0.2 and the λ in equation 9 as 0.1. We set the batch size as 4 and the gradient accumulation steps as 4.

B.3 Additional Details

In data preparation stage, we split the retrieved knowledge chunks into sentences by open-source tools[7] and drop the sentences which is incomplete and too short (less than 5 words). In training stage, we use Adam (Kingma and Ba, 2014) as the optimizer. In inference stage, we use the vllm (Kwon et al., 2023) to accelerate text generation. In text generation process, we use the greedy decoding. All the process, including data preparation, model training and model inference is accomplished on a 8*A800 GPU server.

C Case study

The full content of the case 7 is listed in table 8.

D Prompts For Data Preparation and Inference

In data preparation, training (stage 2) and inference, we use the same prompts. When the input knowledge is none, we use the following prompt:

[7] https://github.com/mediacloud/sentence-splitter

"*Answer the subsequent question. \n\n Question: [Q]* \n\n Answer: ". When the input knowledge is not none, we use the following prompt:*Given the following passage, answer the subsequent question.\n\n Passages: [P] \n\n Question: [Q] \n\n Answer:*. The "[P]" represents the input knowledge and "[Q]" indicates the input question.

Question: Who plays the science officer on Star Trek discovery who is a chief engineer?
Standard answer: "Anthony Rapp", "Anthony Deane Rapp"
Silver context: 1: Yelchin died in a car accident on June 19, 2016, 2: After the first season concluded with the "Discovery" receiving a distress call from the USS "Enterprise", specifically from Captain Christopher Pike, Harberts expressed interest in exploring that character; Anson Mount was **LLM's prediction:** The character who plays the science officer and chief engineer on Star Trek: Discovery is Captain Christopher Pike, portrayed by Anson Mount.
KO-RAG context: 1: Yelchin died in a car accident on June 19, 2016, 2. Science Officers Saru and Stamets, respectively. 3. After the first season concluded with the "Discovery" receiving a distress call from the USS "Enterprise", specifically from Captain Christopher Pike, Harberts expressed interest in exploring that character; Anson Mount was 4: Scotty joins Kirk and Pavel Chekov (Walter Koenig) 5: Scotty joins Kirk's crew aboard the USS "Enterprise"-A at the end of "" (1986). 6. However, the executive producers of the series, Cruz, and GLAAD immediately released a statement saying "death is not always final in the "Star Trek" universe" and that the relationship between Culber and Stamets would continue to be explored. **LLM's prediction:** Based on the passage, the character who plays the science officer and chief engineer on Star Trek: Discovery is Paul Stamets, played by actor Anthony Rapp.

Table 8: Case study