RPDR: A Round-trip Prediction-Based Data Augmentation Framework for Long-Tail Question Answering

Yiming Zhang[1*] **Siyue Zhang**[2] **Junbo Zhao**[1] **Chen Zhao**[3,4]

[1]Zhejiang University [2]Nanyang Technological University [3] NYU Shanghai
[4] Center for Data Science, New York University

Abstract

Long-tail question answering presents significant challenges for large language models (LLMs) due to their limited ability to acquire and accurately recall less common knowledge. Retrieval-augmented generation (RAG) systems have shown great promise in mitigating this limitation by integrating external retrieval mechanisms. However, dense retrieval models often face the same difficulties when generalizing to rare or niche knowledge. In this study, we introduce RPDR, a novel data augmentation framework that selects high-quality easy-to-learn training data, to enhance dense retrievers. Our approach is built around three core components: synthetic data generation, data selection with Round-Trip prediction to identify easy-to-learn instances, and retriever training with these instances. We evaluate RPDR on two long-tail retrieval benchmarks, PopQA and EntityQuestions, demonstrating substantial improvements over existing retrievers like BM25 and Contriver, especially on extremely long-tail categories. We identify the strengths and limitations of RPDR through detailed human analysis and propose a dynamic routing mechanism to dynamically route queries to specialized retrieval modules to further improve retrieval performance.[1]

1 Introduction

The development of large language models (LLMs) has transformed a wide range of applications, from answering questions to supporting complex decision-making (Wu et al., 2024). Despite their impressive capabilities, a key challenge arises when generalizing these models to real-world scenarios: handling long-tail user queries (Kandpal et al., 2023; Dai et al., 2023). For instance, a user might ask about a lesser-known football player as shown in Figure 1. Addressing long-tail question answering (LTQA) is critical because it directly influences user trust and engagement. When systems fail to provide accurate answers to these uncommon queries, users may lose confidence in the system, reducing their likelihood of asking similar questions in the future. This interaction creates a negative feedback loop resulting in a reduction in long-tail training data that further degrades the model's performance.

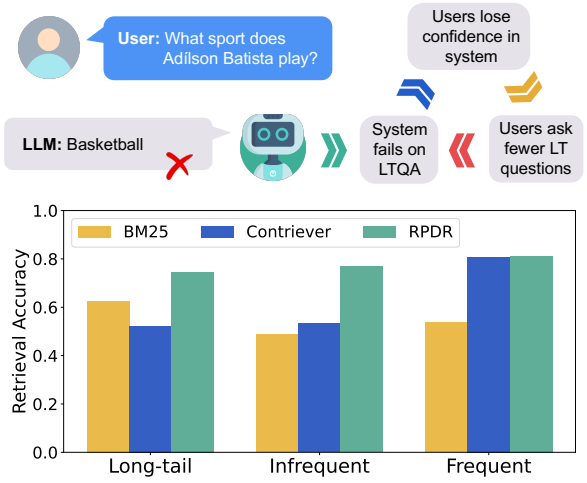

Figure 1: (Top) Negative feedback loop in long-tail question answering in large language model-based systems. (Bottom) We challenge existing findings that dense retrievers struggle on long-tail questions, and argue that through appropriate training, dense retrieval based methods RPDR can surpass BM25 on long-tail retrieval.

Given the scale of todays pre-training data and LLMs, it is expected that LLMs can acquire a huge amount of information and answer long-tail questions with their parametric knowledge. However, recent work (Kandpal et al., 2023) has shown that LLMs struggle to learn long-tail knowledge, and are prone to hallucinating in such

[*]Work done when Yiming Zhang was visiting NYU Shanghai. Correspondence: Yiming Zhang (yimingz@zju.edu.cn), Chen Zhao (cz1285@nyu.edu)

[1]Our code will be made available at https://github.com/yiming-zh/RPDR.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22010–22024
November 4-9, 2025

cases. Retrieval-augmented generation (RAG) has emerged as a promising solution for LTQA (Wu et al., 2024). By integrating external knowledge through retrieval mechanisms, RAG systems aim to mitigate the limitations of relying solely on the parametric knowledge of LLMs. Despite their high potential, the technical challenge is how to retrieve relevant knowledge from millions of candidates for long-tail questions. Traditional term-matching-based information retrieval methods such as BM25 (MacAvaney et al., 2020), fail to capture nuanced semantic similarities. Dense retrieval models (Karpukhin et al., 2020; Izacard et al., 2021) address this by encoding both queries and knowledge pieces into a shared embedding space, allowing similarity to be computed through vector comparisons. However, existing work shows that dense retrievers cannot generalize effectively to long-tail queries and perform even worse than traditional retrieval methods (Kandpal et al., 2023; Sciavolino et al., 2021a).

Our study challenges existing findings and claims that with appropriate training, dense retrieval methods can also excel on long-tail queries. Our intuition is to select easy-to-learn queries that are inherently learnable, in which the original text can be easily recovered from learned embeddings. To this end, we propose a novel data augmentation method RPDR as follows: (1) **Synthetic Data Generation**, which generates new question-answer pairs, in which the questions include long-tail entities; (2) **Data Selection with Round-trip Prediction**, which filters and selects easy-to-learn question-answer pairs decided through a round-trip prediction: an off-the-shelf dense retriever (Izacard et al., 2021) produces an embedding, and the data is selected if a trained inverse model (a decoder) (Morris et al., 2023) can reconstruct the instance. (3) **Retriever Training with Augmented Data**, which trains a new dense retriever with selected easy-to-learn data.

We evaluate RPDR on two long-tail question answering datasets: PopQA (Mallen et al., 2023) and EntityQuestions (Sciavolino et al., 2021a). RPDR demonstrates significant improvements over BM25 and dense retrievers across long-tail and frequent query splits. When combined with an LLM as a generator, the improved retrieval leads to higher answer accuracy, achieving 10.9% improvement on long-tail queries.

Our analyses further indicate that using round-trip prediction to select augmented data substantially improves on a specific category that is common in long-tail queries: syntactically simple but semantically rare and nuanced entities (*e.g.,* John XIX and John X), while still struggling on syntactically complex entities (*e.g.,* Ern Noskó). Following these findings, we propose a routing mechanism to dynamically assign questions to different retrieval modules (*i.e.* RPDR or BM25) based on their characteristics. Compared with using only RPDR, the whole framework achieves a 4.6% improvement.

Our contributions are summarized as follows:

- We challenge existing findings on retrieval-augmented generation approaches for long-tail question answering and claim that, when trained with augmented data, dense retrievers ultimately outperform traditional term-matching methods such as BM25.
- We propose a new data augmentation framework RPDR in the long-tail scenario, which selects easy-to-learn data through round-trip predictions.
- We provide a comprehensive study on the strengths and limitations of RPDR and further propose a routing mechanism to mitigate these limitations.

2 Background

In this section, we first define the long-tail question answering task (§2.1), then introduce the baseline retrieval-augmented QA-based systems (§2.2), and finally, discuss the datasets used in our study (§2.3).

2.1 Long-Tail Question Answering

We formulate our task as open-domain question answering (Karpukhin et al., 2020; Chen, 2017), where the goal is to take a question q and predict an answer y. Our focus is on simple factoid questions, which rely on a single fact—represented as a triplet in the form of (subject, relation, object)—to derive the answer. Normally, the question contains the subject and relation (*e.g.,* "what's the capital of the United States?"), and the answer is the corresponding object (*e.g.,* "Washington D.C."). We then define a question as long-tail if the frequency of the subject entity is smaller than a threshold, as follows:

$$f(e_q) \leq \tau, \tag{1}$$

where τ is the threshold distinguishing low-frequency entities from high-frequency ones.

With the development of LLMs, a straightforward approach is to directly ask these questions to an off-the-shelf LLM (*e.g.,* GPT-4o), with the hope that facts are captured in its parametric knowledge. However, answering long-tail questions poses a significant challenge in LLMs. As recent work (Kandpal et al., 2023) has shown even scaled-up LLMs still struggle to memorize long-tail knowledge that rarely appears in the pretraining data. To make matters worse, LLMs are prone to mixing up long-tail facts with more popular ones stored in their parametric knowledge, often resulting in hallucinations when answering these questions (Kang et al., 2024). Our study adopts an alternative approach that tackles limited memorization in parametric knowledge of LLMs, retrieval-augmented generation (RAG), which we detail next.

2.2 Retrieval-Augmented Generation (RAG)

The goal of RAG is to mitigate the limitations in the parametric knowledge of LLMs (Wu et al., 2024). RAG involves a separate retrieval module to find relevant passages from a large corpus (*e.g.,* Wikipedia) and an LLM to generate an answer based on the question and retrieved passages.

Passage retrieval. Retrieval methods are typically categorized into sparse retrieval and dense retrieval. Sparse retrieval methods, like TF-IDF (Das and Chakraborty, 2018) and BM25 (MacAvaney et al., 2020), rely on surface form matching. In contrast, dense retrieval methods encode questions and passages into embeddings, allowing semantic matching.

Specifically, dense retrieval models use PLMs (*e.g.,* BERT (Reimers and Gurevych, 2019)) to encode the question q and the passage p separately using two independent encoders (*i.e.* bi-encoders (Karpukhin et al., 2020)). These models learn a scoring function (*e.g.,* dot product) between question and passage vectors:

$$f(q,p) = \text{sim}(\text{Enc}_Q(q), \text{Enc}_P(p)). \quad (2)$$

Dense retrievers are highly scalable, since passages can be encoded offline, and are efficiently retrieved over maximum inner product search (MIPS) with the question.

For training, dense retriever models are primarily based on the contrastive learning paradigm.

Property	PopQA	EntityQuestions
Knowledge Corpus		
Source	Wikipedia	Wikipedia
Dataset Statistics		
# Relations	16	24
Dataset Size (# Questions)		
# Training Set	NA	176.6k
# Development Set	NA	22.1k
# Test Set	14.3k	22.1k

Table 1: Basic statistics of the PopQA and EntityQuestions used in our experiments. "NA" means no such set.

Specifically, given a positive passage p^+ and a set of negative passages $p_1^-, ..., p_m^-$, we use negative log-likelihood (NLL) loss:

$$L(q, p^+, p_1^-, ..., p_m^-) = \frac{e^{f(q,p^+)}}{e^{f(q,p^+)} + \sum_{j=1}^{m} e^{f(q,p_j^-)}}. \quad (3)$$

Answer generation. The answer generator takes as input the question, as well as the top passages from the retrieval component, and generates the answer. A widely used approach is Fusion-in-Decoder, which fine-tunes an encoder-decoder LLM (e.g., T5). FiD independently encodes each passage and question, then concatenates their representations before passing them to the decoder, as formulated below:

$$y = \text{Dec}([\text{Enc}([q; p_1]); \ldots; \text{Enc}([q; p_k])]), p_k \in \mathcal{D}. \quad (4)$$

With decoder-only LLMs, the support passages and the question are concatenated into a single sequence as follows:

$$y = \text{Dec}([p_1, \ldots, p_k; q])), p_k \in \mathcal{D}. \quad (5)$$

2.3 Dataset

Considering that the long-tail RAG scenario requires two conditions, long-tail knowledge and a supporting corpus, we conducted experiments and analyses on two benchmark datasets, PopQA (Mallen et al., 2023) and EntityQuestions (Sciavolino et al., 2021a), and their statistics are summarized in Table 1.

- **PopQA** is a large-scale long-tail question answering dataset about entities with a wide variety of popularity. It is grounded in Wikidata. Beyond entities, PopQA features a diverse array of question types, covering topics such as people, places, organizations, and events.

- **EntityQuestions** is another long-tail question-answering dataset. Similar to PopQA, it uses Wikipedia hyperlink counts as a proxy for the frequency of entities and samples knowledge triples from Wikidata. EntityQuestions includes seventeen different relation types.

3 Method

This section introduces our proposed framework, RPDR. As mentioned earlier, we focus on enhancing the capability of dense retrievers for long-tail entities—a challenge that significantly impacts the performance of Retrieval-Augmented Generation (RAG) systems. At a high level, as illustrated in Figure 2, RPDR employs a data augmentation approach, generating new synthetic samples containing long-tail entities to train dense retrievers (§3.1). We emphasize that the quality of these augmented samples is crucial, ensuring that their patterns are easier for dense retrievers to learn. We further hypothesize that easy-to-learn data yield embeddings that can be reliably recovered, as the mapping between text and embeddings is inherently learnable. To this end, we introduce a novel round-trip prediction mechanism to identify high-quality, easy-to-learn data. Specifically, we first train a separate inverse embedding model (§3.2). Given a sample, we use an off-the-shelf dense retriever to generate embeddings from the question and select the sample for training a new dense retriever only if its embedding can be accurately reconstructed by the inverse embedding model (§3.3).

3.1 Synthetic Data Generation

We adopt similar mechanisms to PopQA (Mallen et al., 2023) and EntityQuestions (Sciavolino et al., 2021a) to create the pool from Wikipedia for data augmentation. Specifically, we synthesize samples using the knowledge triples from Wikidata. To ensure that the selected samples are in the long-tail distribution, we use the monthly page view count of the Wikipedia page corresponding to the subject of the knowledge triple answer as a measure of each sample's popularity. We only keep triples with popularity below a predefined threshold.[2] Then, for each triple, we apply templates to generate query-answer pairs, where each query is about the subject and relation, and the answer is the corresponding object. Next, we adopt BM25 to retrieve the top 1,000 passages. We only keep the sample if these passages contain the correct answer string. In total, we generated approximately 86k samples for PopQA and approximately 126k samples for EntityQuestions.

[2]In this paper, we set the threshold of popularity at 1e6. It is also worth noting that during data augmentation, we remove triples already used in PopQA while ensuring that the entire process remains invisible to the EntityQuestions test set.

3.2 Training an Inverse Model

The inverse model (Morris et al., 2023) is designed to reconstruct original text from an embedding. Specifically, given an embedding e generated by a specified embedding model, our goal is to recover the corresponding text x. This can be formulated as generating text whose embedding is maximally similar to the ground-truth embedding as:

$$\hat{x} = \arg\max_{x} \cos(\phi(x), e). \tag{6}$$

Given a dataset $\mathcal{D} = \{x_1, \dots x_n\}$, we train an inverse mapping of the encoder ϕ by modeling the conditional distribution of text given their embeddings $p(x \mid e; \theta)$. We achieve this by optimizing θ via maximum likelihood estimation:

$$\theta = \arg\max_{\hat{\theta}} \mathbb{E}_{x \sim \mathcal{D}} \big[p(x \mid \phi(x); \hat{\theta}) \big]. \tag{7}$$

For technical details, readers can refer to Morris et al. (2023) for more information.

3.3 Data Selection with Round-trip Prediction

After training the inverse model, we select easy-to-learn augmented data for training dense retriever models. For each question-answer sample, we first apply an off-the-shelf dense retriever (Izacard et al., 2021) on the question x to get an embedding $e(x)$, then we adopt the trained inverse embedding model for the reconstructed text $\hat{x}$. Specifically, we define a reconstruction quality score $S(x)$ for x as follows:

$$S(x) = 1 - \frac{\|\phi(\hat{x}) - \phi(x)\|^2}{\|\phi(x)\|^2}. \tag{8}$$

A higher value of $S(x)$ indicates better reconstruction quality. We then select the top k samples in an augmented set D_{selected}, where the reconstruction score $S(x)$ is above a threshold θ:

$$D_{\text{selected}} = \{x \mid S(x) \geq \theta,\ x \in D_{\text{aug}}\}. \tag{9}$$

With the filtered dataset D_{selected}, we use the contrastive learning objective mentioned in Equation (3) to fine-tune the off-the-shelf dense retriever, Contriver (Izacard et al., 2021).

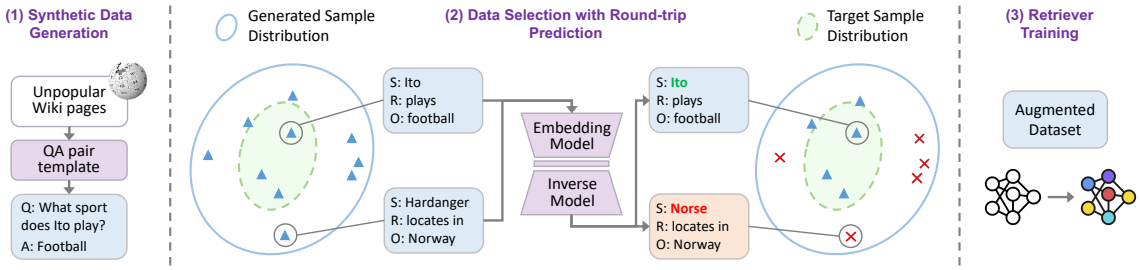

Figure 2: The RPDR framework consists of three main stages: (1) Synthetic Data Generation that generates a pool of long-tail QA pairs. (2) Data Selection with Round-trip Prediction that trains an inverse model to select easy-to-learn samples with reversibility. (3) Retriever Training that trains a dense retriever using the augmented samples.

4 Experiment

In this section, we evaluate RPDR and baseline systems on PopQA and EntityQuestions.

4.1 Experimental Setup

Baselines. We selected two basic retrieval methods as baselines for experimental comparison: the text-based approach BM25 (MacAvaney et al., 2020) and the widely used embedding-based approach Contriever (Izacard et al., 2021). Besides, we selected three state-of-the-art embedding-based retriever models, BGE (Chen et al., 2023b), gemma (Gemma et al., 2024), and NV-Embed (Lee et al., 2024). We include details in Appendix A.

Metrics. Following (Mallen et al., 2023; Wang, 2025; Yu et al., 2024), for retrieval evaluation, we mainly use Retrieval Accuracy, denoted as **R@k**, which measures whether at least one answer string appears in the top-k retrieved passages. For answer evaluation, we use **Exact Match**, which measures the exact string match between the predicted and the gold answer.

To provide a systematic study, we partition the PopQA and EntityQuestions test set with different entity frequencies. Specifically, we split dataset into three categories: **Long-Tail** that represents queries with corresponding entity frequencies between 10 and 100, capturing extreme rare cases; **Infrequent** that corresponds to entity frequencies between 100 and 10,000, offering a moderately rare queries; **Frequent** that with frequencies exceeding 10,000, correspond to high-frequency queries involving well-known entities.

Training Details. We follow Morris et al. (2023)'s setting to train inverse models. We use an off-the-shelf Contriever as an encoder and train the T5-base model on MS MARCO dataset (Bajaj et al., 2018), with a batch size of 128, a maximum of 100 epochs with early stopping, and the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 1e-3. For RPDR, we fine-tune the Contriver model using the selected samples, with a batch size of 32 for 15 epochs, and the AdamW optimizer with an initial learning rate of 5e-6. For PopQA, we include 22k augmented samples as training data. For EntityQuestions, we combine 41K augmented samples with its original training set (176.6K) as our training data. We prompt LLMs with 15 examples for answer generation, following Mallen et al. (2023).

4.2 Main Results

BM25 excels on long-tail queries while dense retrievers outperform other types. Consistent with previous work (Mallen et al., 2023; Sciavolino et al., 2021a), as shown in Table 2, we find that BM25 outperforms all dense retrievers on long-tail queries. This is because dense retrieval models struggle to generalize to represent long-tail entities. As expected, dense retrievers outperform BM25 in frequent and infrequent queries.

RPDR significantly enhances long-tail retrieval. According to Table 2, RPDR significantly outperforms all other methods on long-tail queries, highlighting the crucial role of data augmentation. For example, in the long-tail category of the PopQA dataset, RPDR achieves a 19.5% improvement in R@10 over NV-Embed, the best-performing embedding model trained from larger decoder-only LLMs. **Notably, with data augmentation, RPDR also surpasses BM25 by 11.9% in R@10, further supporting our claim that, with appropriate training, dense retrieval methods can excel in long-tail query scenar-**

Method	PopQA						EntityQuestions					
	Long-tail		Infrequent		Frequent		Long-tail		Infrequent		Frequent	
	R@10	R@20	R@10	R@20	R@10	R@20	R@10	R@20	R@10	R@20	R@10	R@20
BM 25	62.6	66.8	46.2	52.8	53.8	61.1	48.3	63.5	41.9	66.7	55.0	69.2
Contriever	52.3	59.3	53.3	61.1	80.0	85.8	42.9	53.3	53.4	61.0	57.3	67.6
BGE	53.9	61.2	57.4	65.3	82.4	87.9	44.7	60.3	55.1	69.8	60.9	75.2
NV-Embed	55.0	62.3	63.5	70.6	**89.9**	**90.6**	48.4	61.6	57.1	70.7	**71.5**	**84.3**
Gemma	53.8	61.7	66.9	69.3	88.7	90.2	49.3	61.1	**58.2**	**71.5**	70.5	82.7
RPDR	**74.5**	**75.7**	**76.8**	**78.2**	81.3	87.1	**54.6**	**68.1**	57.4	71.2	62.7	70.0
RPDR-Random	66.8	70.0	70.0	74.5	82.7	87.3	51.9	67.7	52.4	69.5	55.9	70.8

Table 2: Comparison of retrieval performance between RPDR and baseline retrievers for questions with long-tail, infrequent, and frequent entities in the PopQA and EntityQuestions datasets. "R@k" means Retrieval Accuracy in the top-k retrieved passages. "RPDR-random" is a model trained with randomly sampled data from the augmented long-tail distribution. More details refer to Appendix B.

ios. On frequent categories, as expected, larger embedding models achieve better performance, while RPDR also slightly outperforms Contriever, which is also initialized from a BERT model. [3]

Data augmentation with round-trip prediction is a key factor. The comparison between RPDR and RPDR-Random in Table 2 indicates the benefits of round-trip prediction-based data augmentation. For instance, in the long-tail category of PopQA, RPDR outperforms RPDR-Random by 7.7%. These improvements validate our intuition that the proposed data selection scheme effectively identifies easy-to-learn data, resulting in more efficient learning.

The improvements in retrieval propagate to QA accuracy. According to Table 3, relying solely on the parametric knowledge of LLMs proves ineffective for long-tail queries. Retrieval augmentation mitigates this issue, and with the superior retrieval capabilities of RPDR, answer accuracy improves significantly over baseline methods. For instance, when using GPT-3.5 as a generator, RPDR surpasses Contriever as a retriever by 11.7%.

4.3 Ablation Study

RPDR augments training data only when both the question and the answer are reconstructed successfully. To analyze the impact of different data selection criteria, we conduct an ablation study. Specifically, based on the reconstruction accuracy of the question entity (Q) and answer entity (A), we categorize the data into three groups and train retrievers accordingly: (1) Q correct & A correct, (2) Q correct & A wrong, and (4) Q wrong & A wrong.[4]

Training with incorrectly recovered samples hurts performance. According to Table 4, training on samples that cannot be recovered (including those with incorrect Q or A) negatively impacts retrieval performance. We hypothesize that training on long-tail queries inherently disrupts the patterns previously learned by the model. In contrast, samples that can be effectively recovered are easier to learn and help mitigate catastrophic forgetting (Khalifa et al., 2020; Korbak et al., 2022).

5 Qualitative Analysis of RPDR and A Routing Mechanism

This section first conducts qualitative analysis on the strengths and limitations of RPDR (§5.1), motivated by these findings, we propose a routing mechanism that aggregates the strengths of multiple retrievers (§5.2).

5.1 Qualitative Analysis

We qualitatively assess the strengths and weaknesses of RPDR by manually reviewing 50 examples where RPDR retrieves the correct passage while Contriever does not, as well as 50 cases where RPDR fails to locate the correct passage. Additional examples are provided in Appendix G.

[3] Due to limited computational resources, we initialized RPDR from a Contriever model. We anticipate that starting from decoder-only LLMs (e.g., LLaMA) would yield additional improvements on both long-tail and other categories, which we leave for future work.

[4] During data generation, we ensure that the query entity has low frequency while the answer does not. As a result, there are very few samples for the "Q correct & A wrong" category, so we exclude it from our study.

Generator	Retriever	PopQA			EntityQuestions		
		Long-tail	Infrequent	Frequent	Long-tail	Infrequent	Frequent
GPT-j-6B	Parametric	11.3	14.4	18.2	9.7	12.1	12.9
	BM25	24.9	19.4	24.5	23.2	27.1	29.8
	Contriever	19.2	24.7	39.1	16.5	26.6	30.2
	RPDR	30.2	29.7	40.0	28.9	32.8	29.2
GPT-neox 20B	Parametric	13.5	21.2	28.9	10.1	12.8	14.8
	BM25	29.1	27.4	29.2	27.8	29.9	35.6
	Contriever	25.6	33.2	40.7	23.7	24.9	34.4
	RPDR	34.9	35.5	40.0	33.5	33.7	35.9
LLaMA-3 8B	Parametric	11.9	21.6	29.9	10.3	14.5	21.7
	BM25	24.7	29.2	30.1	27.8	29.6	32.8
	Contriever	23.7	34.9	39.1	21.8	25.4	30.2
	RPDR	29.8	34.7	41.8	30.1	31.6	33.9
GPT-3.5	Parametric	22.7	36.4	50.3	20.6	29.9	36.8
	BM25	30.5	31.8	42.6	35.9	35.4	35.6
	Contriever	29.7	41.5	52.7	32.5	37.8	38.4
	RPDR	41.4	42.9	52.1	36.4	38.6	38.7

Table 3: End-to-end QA exact match accuracy comparison for RAG systems with different retriever and reader models on the PopQA and the EntityQuestions datasets. "Parametric" means the generator models answer the query by their parametric knowledge.

Category	Long-tail		Infrequent		Frequent	
	R@10	R@20	R@10	R@20	R@10	R@20
Q correct & A correct	74.5	75.7	76.8	78.2	81.3	87.1
Q wrong & A correct	66.8	74.9	68.6	76.6	77.6	85.4
Q wrong & A wrong	62.9	69.8	53.9	62.0	65.8	73.8
RPDR-Random	66.8	70.0	70.0	74.5	82.7	87.3

Table 4: Retrieval performance on the PopQA dataset for retrievers trained with different categories of augmented data. "Q" means the question entity, while "A" means the answer entity. "correct"/"wrong" means the entity can be recovered correctly/wrongly.

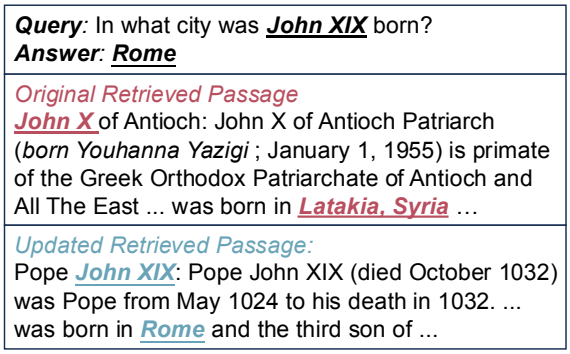

Figure 3: Case study of "John XIX". "Original" means the top-1 retrieved passage from the off-the-shelf Contriever, while "Updated" means the one from RPDR.

RPDR represents nuanced subword differences in long-tail entities. As shown in Figure 3, many long-tail entities differ only by small subword variations, such as John XIX and John X. Dense retrievers struggle with these cases because such entities are rare in the training data, causing them to primarily focus on matching the common word "John". By leveraging data augmentation with round-trip prediction, RPDR learns these nuanced patterns and encodes the distinctions.

RPDR (still) struggles with long-tail entities that have complex syntactic structures. Through error analysis, we find that RPDR cannot embed syntactically complex entities, in 72% of error cases. These entities (*e.g.,* Ern Noskó) often contain complex morphological structures or rare characters, making them harder to learn, and are not included as augmented data (as our inverse embeddings also fail on them). We argue that in such cases, term-matching approaches like BM25 are more effective.

Questions in these datasets are ambiguous. We find that 24% of errors stem from question ambiguity. For example, entities like Finale can refer to either an album or a song, leading to multiple correct answers to the question, "Who is the creator of Finale?"—yet the labels only include Madeon. Additionally, 4% of errors involve equivalent answers (Si et al., 2021). For instance, the retrieved content may contain the word "novel" while the expected answer is "fiction".

5.2 Routing Mechanism

Through qualitative analysis, we identify clear patterns in RPDR's strengths and limitations. Based on these insights, we propose a routing mechanism that dynamically switches between two retrieval strategies—BM25 and RPDR—depending on the characteristics of the input query (Mallen et al., 2023). Specifically, we formulate this as a binary classification problem, where the input is the query, and the output is a binary label that determines which retriever to route to.

Setup. We fine-tune a Sentence-BERT (Reimers and Gurevych, 2019) model for classification. We randomly select 10,000 augmented samples as training data, with half correctly predicted by Contriever and the other half by BM25. We use a batch size of 32 for 5 epochs with the AdamW optimizer and an initial learning rate of 2e-5.

Results. As shown in Table 5, our routing mechanism further improves retrieval performance, particularly for long-tail and infrequent categories. For instance, on PopQA, RPDR + RM achieves a 4.6% improvement in R@10 over RPDR. These results validate our hypothesis that combining the strengths of different retrievers is beneficial.

PopQA						
Method	**Long-tail**		**Infrequency**		**Frequency**	
	R@10	R@20	R@10	R@20	R@10	R@20
RPDR	74.5	75.7	76.8	78.2	81.3	87.1
RPDR + RM	79.1	80.3	78.7	80.0	81.3	87.9
EntityQuestions						
Method	**Long-tail**		**Infrequency**		**Frequency**	
	R@10	R@20	R@10	R@20	R@10	R@20
RPDR	54.6	68.1	57.4	71.2	62.7	70.0
RPDR + RM	58.4	72.2	59.5	71.3	61.8	71.1

Table 5: Retrieval performance comparison of RPDR without and with the routing mechanism on the PopQA and the EntityQuestions datasets.

6 Related Work

LLM hallucinations. Large Language Models are susceptible to hallucinations, generating plausible yet incorrect content (Niu et al., 2024). To address this, research has focused on retrieval-augmented methods that ground outputs in external knowledge sources (Asai et al., 2023; Siriwardhana et al., 2023; Reichman and Heck, 2024). Our work builds on retrieval-augmented methods, seeking to enhance the generalization capabilities of LLMs in long-tail scenarios where existing methods still falter.

Dense retrieval models. The retrieval model plays an important role in the retrieval-augmented generation (Fan et al., 2024). Dense retrieval models encode both queries and documents into dense vectors using transformer-based models (Karpukhin et al., 2020; Izacard et al., 2021; Xiong et al., 2021; Gao et al., 2022; Yu et al., 2022). These models outperform sparse retrieval methods like BM25 (Robertson et al., 2009), by capturing semantic similarity more effectively. Recent studies have focused on distilling or fine-tuning LLM embeddings and demonstrated outstanding performance (Chen et al., 2023b; Gemma et al., 2024; Hu et al., 2024; Lee et al., 2024). We focus on improving the dense retriever with data augmentation for long-tail question answering, where their performance used to lag behind.

Data augmentation. Data augmentation (Chen et al., 2023a) has emerged as a critical technique for addressing data scarcity in the NLP community. There are various data augmentation approaches, from token level such as synonym replacement (Kolomiyets et al., 2011) to sentence level like back-translation (Edunov et al., 2018). Recent approaches mainly sample LLMs to get augmented data (Ding et al., 2024). We propose a round-trip prediction to select high-quality and easy-to-learn augmented data for long-tail question answering.

7 Conclusion

In this work, we challenge existing findings on long-tail QA by arguing that dense retrieval methods, when trained with high-quality and easy-to-learn augmented data, can outperform traditional term-matching approaches such as BM25. Our proposed RPDR framework enhances retrieval quality by adopting synthetic data generation and a novel round-trip prediction mechanism, leading to significant improvements on benchmark datasets and boosting overall end-to-end system performance. Furthermore, our analysis sheds light on the strengths and limitations of RPDR, and we introduce a routing mechanism that combines the advantages of multiple retrievers, offering additional performance enhancements.

8 Limitations

First, our round-trip prediction framework for selecting easy-to-learn augmented data incurs additional computational costs during data construction and model fine-tuning. Second, existing long-tail QA datasets primarily focus on single-fact questions, leaving the extension of our findings to long-tail questions requiring complex reasoning (e.g., multi-hop questions) largely unexplored. Third, while our work mainly focuses on short-form question answering, future research could extend this approach to long-form generation tasks that necessitate long-tail knowledge as part of the process. At last, some valuable datasets, such as Head-to-Tail (Sun et al., 2023), cannot serve as our experimental settings either due to the lack of an associated corpus or because they do not reflect the long-tail scenarios that our work targets.

9 Acknowledgement

This paper is supported by the National Regional Innovationand Development Joint Fund (No. U24A20254). Junbo Zhao is also supported by the NSFC under Grants (No. 62402424) and the Fundamental Research Funds for the Zhejiang Provincial Universities (226-2024-00049). Chen Zhao is supported by NYU Shanghai Center for Data Science.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

P Bajaj, D Campos, N Craswell, L Deng, J Gao, X Liu, R Majumder, A McNamara, B Mitra, T Nguyen, et al. 2018. A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

D Chen. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023a. An empirical survey of data augmentation for limited data learning in nlp. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023b. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

Yi Dai, Hao Lang, Yinhe Zheng, Fei Huang, and Yongbin Li. 2023. Long-tailed question answering in an open world.

Bijoyan Das and Sarit Chakraborty. 2018. An improved text sentiment classification model using tf-idf and next word negation.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Luu Anh Tuan, and Shafiq Joty. 2024. Data augmentation using llms: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1679–1705.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6491–6501.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2022. Hybrid retrieval models for question answering. *arXiv preprint arXiv:2204.14242*.

Gemma, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol

Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar finetuning examples control how language models hallucinate. *arXiv preprint arXiv:2403.05612*.

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. *arXiv preprint arXiv:2004.04906*.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. 2020. A distributional approach to controlled text generation. *arXiv preprint arXiv:2012.11635*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, volume 2, pages 271–276. ACL; East Stroudsburg, PA.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv preprint arXiv:2405.17428*.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

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)*.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, KaShun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Benjamin Reichman and Larry Heck. 2024. Dense passage retrieval: Is it retrieving? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13540–13553.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021a. Simple entity-centric questions challenge dense retrievers. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021b. Simple entity-centric questions challenge dense retrievers. *arXiv preprint arXiv:2109.08535*.

Hassan S Shavarani and Anoop Sarkar. 2024. Entity retrieval for answering entity-centric questions. *arXiv preprint arXiv:2408.02795*.

Chenglei Si, Chen Zhao, and Jordan Boyd-Graber. 2021. Whats in a name? answer equivalence for open-domain question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9623–9629.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-tail: how knowledgeable are large language models (llms)? aka will llms replace knowledge graphs? *arXiv preprint arXiv:2308.10168*.

Jingjin Wang. 2025. Proprag: Guiding retrieval with beam search over proposition paths. *arXiv preprint arXiv:2504.18070*.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. 2024. Retrieval-augmented generation for natural language processing: A survey.

Lee Xiong, Chenyan Yu, Sharan Chang, Jialu Yu, Qifa Guo, Huizhong Sun, Hao Cheng, Tom Kwiatkowski, Kristina Toutanova, Michael Collins, et al. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2010.02666*.

Chenyan Yu, Lee Xiong, and Jamie Callan. 2022. A survey on conversational dense retrieval: Methods and challenges. *arXiv preprint arXiv:2201.08452*.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184.

A Details of Main Experiments

Baselines. We selected two basic retrieval methods as baselines for experimental comparison: the text-based approach BM25 (MacAvaney et al., 2020) and the widely used embedding-based approach Contriever (Izacard et al., 2021). **BM25** is a probabilistic ranking model scoring documents based on term frequency, inverse document frequency, and document length normalization. **Contriever**, more precisely, the off-the-shelf Contriever is a dense retrieval method that leverages unsupervised contrastive learning to build vector representations for queries and documents. Contriever excels in zero-shot and domain-agnostic scenarios due to its robust pretraining approach.

For better comparison, we also introduce several state-of-the-art embedding-based retriever models, i.e. BGE (Chen et al., 2023b)[5], gemma (Gemma et al., 2024)[6], NV-Embed (Lee et al., 2024)[7]. These models are widely recognized for their superior performance. The primary performance advantages of these models stem from their training on more extensive and diverse datasets.

B More Details about Random Selection

As mentioned earlier, RPDR-random is a model trained with randomly sampled data of the same size as RPDR, drawn from the same long-tail portion and using the same training hyperparameters. In this section, we additionally report its performance in the end-to-end setting. Furthermore, we introduce another baseline, denoted as Full-random, where the same number of training examples are randomly sampled from the full distribution (i.e., not restricted to the long-tail) (Mallen et al., 2023; Sciavolino et al., 2021a) and used to train a model under the same setup.

The results in Table 6 show that sampling from the full distribution does not effectively enhance RAG performance in the long-tail setting; its performance is instead more similar to the original Contriever model. Moreover, performing random selection directly on long-tail data does not lead to performance improvements, which is consistent with the retriever's standalone results. In contrast, our proposed RPDR data selection strategy better preserves the model's performance under the long-tail distribution, aligning with our prior hypothesis.

[5]https://huggingface.co/BAAI/bge-m3

[6]https://huggingface.co/google/gemma-2-2b-it

[7]https://huggingface.co/nvidia/NV-Embed-v2

C Data Augmentation Scale and Model Performance

We conduct an experiment on the relationship between data augmentation size and model performance. According to Figure 4, the performance gradually increases with more augmented data, as expected.

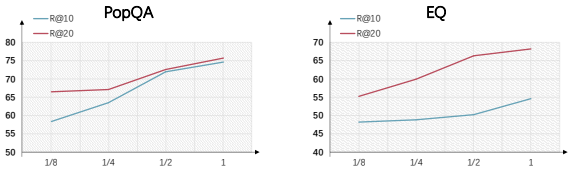

Figure 4: The Relationship between augmented data scale and retriever model performance. In the left figure, EQ represents ENTITYQUESTIONS. The x-axis indicates the proportion of the dataset size relative to the full augmentated dataset.

D Main Results evaluated using IR Metric

In the main body of this paper, we adopt the evaluation metric commonly used in prior work (Sciavolino et al., 2021b; Wang, 2025; Yu et al., 2024; Mallen et al., 2023), **Exact Match**, which checks whether the gold answer appears in the retrieved passage. In this section, we additionally report standard metrics from the information retrieval (IR) domain. Specifically, we treat the first paragraph of the corresponding Wikipedia entity as the gold passage, following Shavarani and Sarkar (2024).

The detailed results are shown in Table 7. It can be clearly observed that the conclusion, "RPDR significantly enhances long-tail retrieval", drawn from our main results, still holds under the modified metric. We also observe that dense retrievers tend to perform better in retrieving the gold passage, which may be attributed to the strong alignment between the training objective and the retrieval task.

However, we emphasize that the gold passages used in this evaluation were automatically constructed by extracting the first paragraph of the corresponding Wikipedia entity using scripts, without human annotation. Therefore, the results in this table should be considered for reference only.

Generator	Retriever	PopQA			EntityQuestions		
		Long-tail	Infrequent	Frequent	Long-tail	Infrequent	Frequent
LLaMA-3 8B	Contriever	23.7	34.9	39.1	21.8	25.4	30.2
	FULL-random	23.3	34.9	42.1	22.6	27.8	33.7
	RPDR-random	25.2	32.5	42.0	27.4	27.1	32.4
	RPDR	29.8	34.7	41.8	30.1	31.6	33.9

Table 6: End-to-end QA exact match accuracy comparison for RAG systems with different retriever and LLaMA-3 8B on the PopQA and the EntityQuestions datasets. "FULL-random" means the retriever model trained on the examples randomly sampled from the same distribution as the data source. "RPDR-random" means the retriever model trained on the examples randomly sampled from the same long-tail distribution as RPDR.

Method	PopQA			EntityQuestions		
	Long-tail Recall@10	**Infrequent** Recall@10	**Frequent** Recall@10	**Long-tail** Recall@10	**Infrequent** Recall@10	**Frequent** Recall@10
BM 25	54.3	48.5	66.9	41.7	51.3	60.8
Contriever	61.2	67.4	87.3	56.9	63.2	73.1
BGE	66.4	77.5	89.7	61.1	69.8	77.1
NV-Embed	71.8	**80.8**	**94.5**	67.3	75.2	**81.9**
Gemma	67.6	75.3	92.5	67.4	73.2	79.0
RPDR	**74.3**	79.1	88.3	**72.6**	**75.4**	78.7
RPDR-Random	69.2	73.6	88.9	68.5	74.3	77.4

Table 7: Comparison of retrieval performance between RPDR and baseline retrievers for questions with long- tail, infrequent, and frequent entities in the POPQA and ENTITYQUESTIONS datasets. "Recall@10" means the gold passage being in the top-10 recall results.

Model	Long-tail	Infrequent	Frequent
Contriever	52.3	53.3	80.0
RPDR-random	66.8	70.0	82.7
RPDR (Q wrong & A wrong)	62.9	53.9	65.8
RPDR-trained with full dataset	72.1	71.9	73.4
RPDR	74.5	76.8	81.3

Table 8: Comparison across different training data.

E Full Dataset vs. Data After Filter

Considering that the full dataset contains a larger volume of data, in this section we will discuss the necessity of the data filtering step. We report R@10 retrieval performance on PopQA in Table 8. The results show that training on the full set of 86k samples leads to suboptimal retrieval performance across long-tail, infrequent, and frequent queries. We suspect that the presence of noisy and low-quality synthetic data in the full dataset contributes to representation drift. Moreover, filtering reduces the training data size from 86k to 22k, significantly lowering computational costs. By comparing RPDR with baselines trained on randomly sampled data, wrongly generated data, and the full dataset, we confirm that RPDR offers advantages in both higher accuracy and lower cost.

F Scaling Dynamics of the Training Dataset over Multi-cycle Round Trips

Would a second round-trip filtering cycle help the already fine-tuned model? We ran the procedure again and obtained fewer than 1k additional examplestoo small compared with the original 22k poolso we chose not to retrain the fine-tuned model with this marginal increment. Table 9 shows the variation in training data size across multiple cycles. Given the minor variation in scale, we refrained from extending the experiments to the second and third rounds.

	Cycle 1	Cycle 2	Cycle 3
Data scale	22k	23k	23k

Table 9: Comparison across different training data.

G The Samples for Qualitative Snalysis

As mentioned in Section 5.1, we analyzed some correct and error case for qualitatively assessing the strengths and weaknesses of RPDR. In this part, we will present the specific samples that could not be included in the main text due to space limitations in Table 10.

Hard-to-embed means syntactically complex entities. These entities (*e.g.,* Ern Noskó often con-

tain complex morphological structures or rare characters, making them harder to learn. Moreover, they are not included as augmented data (as our inverse embeddings also fail on them).

Semantic Ambiguity means entities with multiple meanings tend to be interpreted by the model based on their more frequently used semantics, causing it to over-prioritize common but incorrect matches. For instance, terms like *Milk* and *Finale*, which have multiple plausible interpretations, are often associated with the retriever ranking popular but unrelated results above the correct but less common ones.

Unanswerable Queries means the query has no answer in the retrieved passages even though the description is truly related to the target entity. This type of error is primarily caused by issues in the design of sample answers.

Error Type	Query	Answer	Retrieved Passage
Hard-to-Embed Target Entities	What is *Edwin Wallock*'s occupation?	actor, actress	Charles Edwin (died 1756): The Government could field no candidates at the rerun of the election and Edwin was returned unopposed as Member of Parliament (MP) for Westminster ... Charles Edwin (died 1756)
Semantic Ambiguity	What genre is *Milk*?	biographical film, biopic	Milk: ... Freezing of milk can cause fat globule aggregation upon thawing, resulting in milky layers and butterfat lumps. ... Milk is often served in coffee and tea. Steamed milk is used to prepare espresso-based drinks such as cafe latte.
Semantic Ambiguity	Who was the director of *Finale*?	Ken Kwapis, Kenneth William Kwapis	Finale (software): Finale is the flagship program of a series of proprietary music notation software developed and released by MakeMusic for the Microsoft Windows and macOS operating systems. ... including the score for an entire ensemble (e.g., orchestra, concert band, big band, etc.) and parts for the individual musicians.
Unanswerable Queries	What genre is *Great Expectations*?	fiction, fictional	Great Expectations: The film adaptation in 1946 gained the greatest acclaim... Following are highlights of the adaptations for film and television, and for the stage, since the early 20th century.

Table 10: Three major categories of RPDR errors.