Training a Utility-based Retriever Through Shared Context Attribution for Retrieval-Augmented Language Models

Yilong Xu[1,2,3] **Jinhua Gao**[1,2*] **Xiaoming Yu**[1,2] **Yuanhai Xue**[1,2] **Baolong Bi**[1,2,3]
Huawei Shen[1,2,3] **Xueqi Cheng**[1,2,3]
[1]State Key Lab of AI Safety, Institute of Computing Technology, CAS
[2]Key Lab of AI Safety, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
{xuyilong23s, gaojinhua}@ict.ac.cn

Abstract

Retrieval-Augmented Language Models boost task performance, owing to the retriever that provides external knowledge. Although crucial, the retriever primarily focuses on semantics relevance, which may not always be effective for generation. Thus, utility-based retrieval has emerged as a promising topic, prioritizing passages that provide valid benefits for downstream tasks. However, due to insufficient understanding, capturing passage utility accurately remains unexplored. This work proposes SCARLet, a framework for training utility-based retrievers in RALMs, which incorporates two key factors, multi-task generalization and inter-passage interaction. First, SCARLet constructs shared context on which training data for various tasks is synthesized. This mitigates semantic bias from context differences, allowing retrievers to focus on learning task-specific utility and generalize across tasks. Next, SCARLet uses a perturbation-based attribution method to estimate passage-level utility for shared context, which reflects interactions between passages and provides more accurate feedback. We evaluate our approach on ten datasets across various tasks, both in-domain and out-of-domain, showing that retrievers trained by SCARLet consistently improve the overall performance of RALMs.

Resources: github.com/ylXuu/SCARLet

1 Introduction

Retrieval-Augmented Language Models (RALMs; Lewis et al., 2020) typically comprise two parts: the retriever and the generator. The retriever collects up-to-date task-related external information, while the generator incorporates the collected non-parametric knowledge into inference. RALMs have achieved enhanced performance across various downstream tasks, including question answering, fact checking, and dialogue generation (Shao et al., 2023; Cheng et al., 2023). As a crucial role, the optimization of the retrievers in RALMs has become a trending research topic.

Early RALMs adopt relevance-based retrievers, including both sparse (Robertson and Zaragoza, 2009) and dense (Karpukhin et al., 2020) models. However, these retrievers are primarily biased toward semantic relevance (Wu et al., 2024), failing to consider the passage utility and leading to misalignment in RALMs. The utility, measuring the valid gain that a passage contributes to the downstream generation (Zhang et al., 2024), can bridge the gap between the retriever and the generator. Some recent works have proposed to optimize retrievers by constructing feedback from generators (Shi et al., 2023; Yu et al., 2023; Wei et al., 2024), achieving promising results. Nonetheless, how to align retrievers to better capture utility remains an open yet challenging problem.

Different from relevance, which is mainly determined by the query and the passage (Xia et al., 2015; Guo et al., 2019), utility needs a more comprehensive measurement. In this paper, we propose the following two vital yet overlooked factors for utility modeling in RALMs:

Multi-task Generalization RALMs need to accommodate various downstream tasks, where the utility of a passage can vary accordingly. Existing methods typically optimize retrievers using the pooling strategy, i.e., mixing data from different tasks for training, to learn task-specific retrieval criteria (Lin et al., 2024; Zamani and Bendersky, 2024). However, since pooled samples from different tasks typically have different contexts, the trained retrievers might tend to capture semantic relevance signals instead of utility features. Such unexpected preference will downgrade the retrievers' generalization ability, especially for those with weaker linguistic capabilities (Liu et al., 2024).

[*]Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 629–648
November 4-9, 2025 ©2025 Association for Computational Linguistics

Inter-passage Interaction In some complex tasks, the utility of a certain passage cannot be solely determined by itself. For example, when handling multi-hop question-answering tasks, the model should rely on preceding and even succeeding contexts in the reasoning chain to judge a passage's utility. However, the utility signals constructed in previous works either fail to provide passage-level feedback (Zamani and Bendersky, 2024; Sohn et al., 2024) or evaluate each passage independently (Yu et al., 2023; Shi et al., 2023), leading to imprecise utility measurements.

In this paper, we propose a novel framework to train utility-based retrievers for RALMs, named **SCARLet**, representing shared context attribution supervised training for utility-based retrievers.

Specifically, SCARLet first introduces a training data synthesis pipeline. Contrary to the previous pooling strategy that mixes training data with different contexts, our pipeline first constructs a shared context, and subsequently synthesizes training data for various downstream tasks derived from the shared context. This method mitigates the semantic interference by achieving single-variable control, and enables the retriever to focus on learning task-specific utility. To better assess the utility of certain passages, SCARLet employs a passage-level perturbation-based technique, which randomly removes some passages from the context and measures the fluctuations in the generated output. Such a design can effectively capture the synergy between passages, thereby accurately reflecting their utility. Finally, SCARLet collects positive and negative samples based on the utility scores and trains the retriever in a contrastive way.

We conduct extensive experiments to evaluate the performance gain brought by SCARLet. Our experiments adopt ten datasets, covering eight distinct tasks that are frequently used for RALMs evaluation. The results show that RALMs equipped with retrievers trained by SCARLet, consistently achieve optimal or suboptimal downstream performance across all datasets. Moreover, further analysis and case studies demonstrate that SCARLet can better capture utility signals.

To summarize, our main contributions include:

- We argue that utility should be preferred in RALMs and propose two critical factors for training utility-based retrievers.
- We propose SCARLet, a novel framework to train utility-based retrievers through shared context synthesis and utility attribution.
- We conduct extensive experiments across various tasks, demonstrating that our proposed SCARLet can improve the overall performance of RALMs.

2 Related Work

RALMs Large Language Models (LLMs; Brown et al., 2020) exhibit remarkable performance across a wide range of tasks (Zhao et al., 2024; Naveed et al., 2024; Wei et al., 2022). However, LLMs also face the challenge of hallucinations, often performing poorly when addressing factual issues (Huang et al., 2024; Bi et al., 2024). The emergence of RALMs effectively alleviates the weakness of insufficient factuality (Gao et al., 2024). A RALM system typically comprise a retriever and a generator, where the retriever recalls external information to enhance the generator to respond more accurately. To further optimize RALMs and improve the synergy between the two parts, existing methods generally fall into three categories: 1) overall optimization (Lin et al., 2024; Zamani and Bendersky, 2024); 2) generator-only optimization (Fang et al., 2024; Yu et al., 2024; Bi et al., 2025); 3) retriever-only optimization (Shi et al., 2023; Yu et al., 2023). Optimizing only the retriever is a more efficient and cost-effective way that offers plug-and-play capabilities, enhancing the overall efficiency and stability of the RALM system.

Utility-based Retrieval In RALMs, early exploration of retrieval utility focuses on capturing the downstream feedback of generators. Salemi and Zamani (2024) propose supervision based on downstream task metrics, but fail to provide passage-level utility feedback. Shi et al. (2023); Yu et al. (2023) assess utility of each passage using generator outputs, but they ignore the interactions between passages. Sohn et al. (2024); Wei et al. (2024) employ the generator's self-reflection to evaluate utility, which may bring hallucinations as the language models can be dishonest (Madsen et al., 2024). Asai et al. (2023); Glass et al. (2022) notice the multi-task nature of the retrieval stage, but fail to account for the training biases introduced by contextual differences in the pooling strategy.

Therefore, our proposed SCARLet framework comprehensively considers the above issues of multi-task generalization and utility assessment, of-

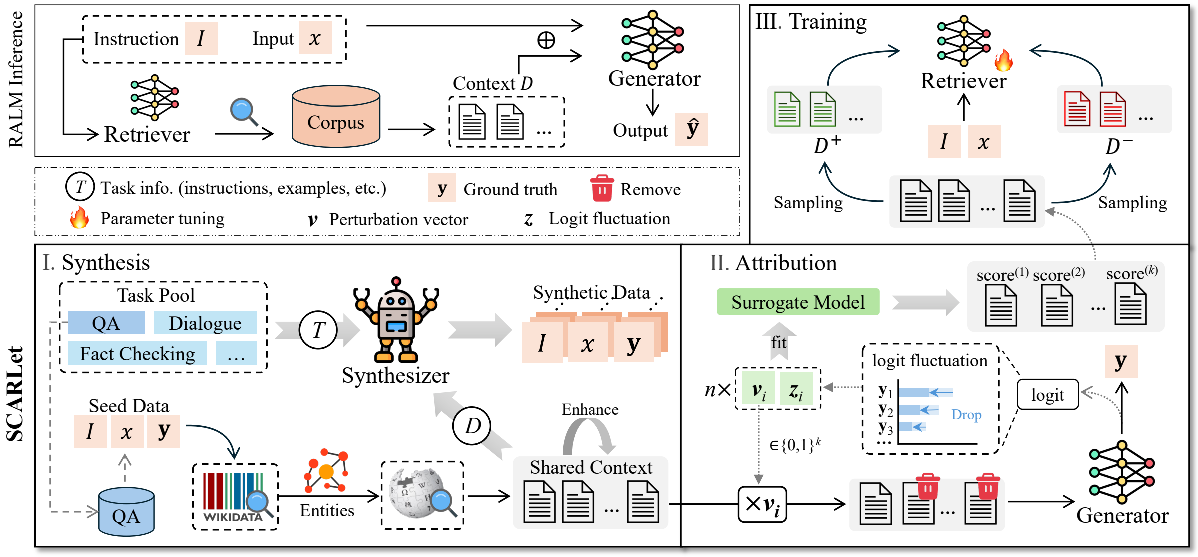


Figure 1: The illustration of SCARLet. The upper left part describes the inference process of RALMs. In SCARLet, there are three main stages. First, the shared context is constructed by retrieving external corpus based on the seed data. The synthesizer is instructed with shared context and task information from the task pool, to generate synthetic data. Next, using the shared context as the data source, SCARLet applies perturbation-based utility attribution on the generator, and then, based on the utility scores, samples positive and negative passages for retriever training.

fering a novel pipeline with shared context synthesis and utility attribution to effectively train utility-based retrievers in RALMs.

3 Method

In this section, we first define the RALMs system, then we introduce the SCARLet pipeline.

3.1 Definitions

A typical RALM system consists of a retriever and a generator. During the retrieval stage, we employ a dense retriever based on an encoder **Enc** with parameters ϕ. And the retriever interacts with an external corpus $\mathcal{C}$. For a query q, we calculate the dot product of the embeddings of q and each passage d in $\mathcal{C}$, as follows:

$$\mathrm{score}\left(q,d\right)=\mathbf{Enc}_{\phi}\left(q\right)\cdot\mathbf{Enc}_{\phi}\left(d\right),d\in\mathcal{C}. \quad (1)$$

The top-k passages with the highest scores are selected and added to the context, denoted as $D=[d_1,\ldots,d_k]$. Note that RALMs need to accommodate various downstream tasks, for a task T and an input x from its dataset, we define the query format as $q=I\oplus x$, where I denotes the instruction description of task T.

In the generation stage, a language model LM with parameters θ serves as the generator. The context D is used to enhance generation, ultimately producing the predicted output $\hat{\mathbf{y}}$, as shown below:

$$\hat{\mathbf{y}}=\mathrm{LM}_{\theta}\left(I\oplus x\oplus D\right), \quad (2)$$

where $\hat{\mathbf{y}}$ is a sequence and $\hat{\mathbf{y}}_t$ denotes the t-th token. We denote the ground truth of x as $\mathbf{y}$.

3.2 Overview of SCARLet

The overall architecture of our proposed SCARLet is shown in Figure 1, including shared context synthesis and training data construction (§3.3), utility attribution modeling (§3.4), as well as data sampling and retriever tuning (§3.5).

Shared context refers to the common context for data of different tasks in the training stage, which is then used to enhance downstream generation. Previous studies employ the pooling strategy (Lin et al., 2024; Zamani and Bendersky, 2024), where each instance has a distinct context for training. Learning task-specific features to improve multi-task generalization of utility-based retrieval might be disturbed by the semantically relevant noise introduced by differences in context, leading to unexpected preference, particularly in retrievers with weaker linguistic capabilities. To tackle the above challenges, our proposed SCARLet adopts a reverse strategy, first constructing shared context to narrow the semantic gap, and then synthesizing task-specific data based on this context. Sharing context across tasks can highlight utility feature differences, making it easier to learn. Moreover, LLM-driven data synthesis has been shown to be a promising way (Long et al., 2024; Kim and Baek, 2025), which can effectively reduce labor costs.

Utility attribution modeling refers to local explanation techniques to build utility signals from the downstream generation. More specifically, we adopt the contributive attribution model, which measures how the input context contributes to the model's output and aligns well with the definition of utility in RALMs. Previous research on optimizing retrievers from downstream generation, either fails to construct passage-level feedback or only considers the individual impact of each passage, overlooking the synergy effects between passages. Therefore, taking the shared context as the source data, SCARLet uses a passage-level perturbation-based utility attribution approach, where fluctuations in generation caused by perturbations can reflect interactions between passages and then be quantified as utility scores.

3.3 Shared Context Synthesis

Specifically, we first define a task pool $\mathcal{T}$, which is linked to various downstream tasks and their datasets, such as multi-hop QA, long-form QA, and fact checking. We begin by collecting seed data from datasets of $\mathcal{T}$, including task instructions, inputs, and ground truth. In line with the motivation behind shared context, passages within this context need to be closely related to facilitate the synthesis of high-quality data. Therefore, we employ an approach based on associated entities, which extracts entities from the seed data, searches their adjacent entities by querying Wikidata[1], and merges them to obtain a related entity list. We then use this list to retrieve relevant passages from the Wikipedia corpus, and treat the recalled passages as the shared context D_{shared}. Subsequently, we instruct the synthesizer model S to generate new training data, using D_{shared} as the information source and task information (including instructions and examples) from $\mathcal{T}$. The process is formalized as follows:

$$\left(x_{T_1}^{\text{new}}, \mathbf{y}_{T_1}^{\text{new}}\right), \ldots, \left(x_{T_l}^{\text{new}}, \mathbf{y}_{T_l}^{\text{new}}\right) = \mathsf{S}\left(D_{\text{shared}}, \mathcal{T}\right), \tag{3}$$

where $x_{T_i}^{\text{new}}$ and $\mathbf{y}_{T_i}^{\text{new}}$ represent input and ground truth of the synthetic data of task T_i, respectively. l is the total number of tasks in $\mathcal{T}$.

To improve the quality of synthetic data, the task pool $\mathcal{T}$ not only provides the task instructions but also offers example data. To further improve robustness, following Fang et al. (2024); Zhang et al. (2024), we also introduce synthetic noise into the shared context by instructing the synthesizer to generate semantically relevant but useless passages. In addition, we incorporate a data filtering step that instructs the synthesizer to eliminate samples containing faults. For more details, please refer to Appendix A. We also provide an example of shared context in Appendix F.

3.4 Passage-level Utility Attribution

Specifically, the context D recalled by the upstream retriever consists of k passages. To evaluate the utility of each individual passage with inter-passage interactions, we adopt a perturbation-based method where we remove certain passages and inspect the changes in the final output. The approach is implemented via introducing a perturbation vector $\mathbf{v} \in \{0,1\}^k$, where 0 and 1 indicate whether the corresponding passage is removed or included, respectively. However, running all generations of 2^k possible perturbation vectors can result in significant computational overhead. Inspired by the method of Local Interpretable Model-agnostic Explanations (LIME; Ribeiro et al., 2016; Mardaoui and Garreau, 2021), we first sample n perturbation vectors randomly and then fit a surrogate model for predicting the utility score, as shown below:

$$\hat{\boldsymbol{\alpha}} \in \underset{\boldsymbol{\alpha} \in \mathbb{R}^{k+1}}{\arg\min} \left\{ \sum_{i=1}^{n} \left(z_i - \boldsymbol{\alpha}^T \mathbf{v}_i\right)^2 + \lambda \left\|\boldsymbol{\alpha}\right\|^2 \right\}, \tag{4}$$

where we adopt the ridge regression (Hilt and Seegrist, 1977) as our surrogate model, $\boldsymbol{\alpha}$ represents the parameters to be fitted, λ is a hyperparameter for regularization, and z_i is the observed value under $\mathbf{v}_i$. More specifically, $\boldsymbol{\alpha}^{(i)}$ denotes the utility score of passage d_i, $\boldsymbol{\alpha}^{(0)}$ represents the intercept term. And z_i, which quantifies the fluctuation caused by $\mathbf{v}_i$, is calculated using the logit values of the tokens in the ground truth $\mathbf{y}$ at each time step, as shown below:

$$z_i = \sum_t \text{logit}\left(\mathbf{y}_t^{(i)}\right). \tag{5}$$

To evaluate the effectiveness of the above utility attribution method, we conduct a preliminary experiment on the GTI benchmark (Zhang et al., 2024), which includes three datasets: HotpotQA (Yang

[1] https://www.wikidata.org

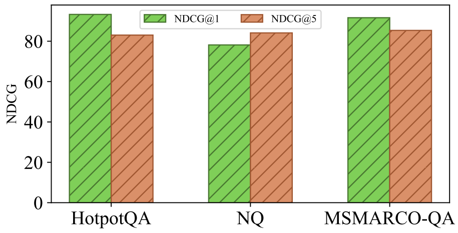


Figure 2: The performance of the perturbation-based attribution method on the GTI benchmark. The nDCG metrics show that it achieves at least about 80% performance on three datasets, with some exceeding 90%.

et al., 2018), Natural Questions (NQ; Kwiatkowski et al., 2019), and MSMARCO-QA (Bajaj et al., 2018). Each test sample comprises input, ground truth, and ten passages including correct passages and other noise passages. We use the utility score to rank the passages. The results, measured using nDCG, demonstrate that our method shows a high accuracy in reflecting passage utility, as shown in Figure 2. We also compare our method to other attribution approaches, and our method outperforms them by over 20%. For further details of the experiment, please refer to Appendix B.

3.5 Sampling and Training

After calculating the utility score for each passage in the shared context, we then collect positive and negative samples based on these scores for training the retriever. When sorted in descending order of the scores, the utility distribution follows an inverse S-shaped curve, as depicted in Figure 3. Passages with higher scores correspond to positive samples, while those with lower scores represent negative samples. To effectively sample these two types of data, we employ a one-dimensional clustering approach. Specifically, we take the utility score list as the input and divide it into three clusters: one for the positive samples, one for the intermediate samples that will be discarded, and another for the negative samples. This method can dynamically adjust the number of useful passages in the context on various tasks and data.

After obtaining positive and negative samples, following Xiong et al. (2020), the loss function is calculated as follows:

$$\mathcal{L} = \sum_{x} \sum_{d^+ \in D^+} \sum_{d^- \in D^-} l\left(\text{score}\left(x, d^+\right), \text{score}\left(x, d^-\right)\right), \quad (6)$$

where l represents the cross-entropy loss.

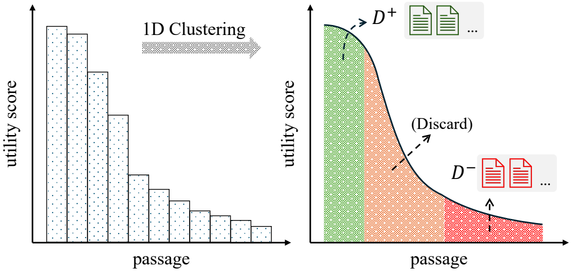


Figure 3: The illustration of the 1D clustering sampling. Based on the utility score, this method clusters the passages into three labels: the high-score passages (green) corresponding to positive samples, the middle-score passages (orange) that will be discarded, and the low-score passages (red) corresponding to negative samples.

Dataset	Task	Corpus	Metric
In-domain			
NQ (Kwiatkowski et al., 2019)	Single-hop QA	Wikipedia	Accuracy
HotpotQA (Yang et al., 2018)	Multi-hop QA	Wikipedia	Accuracy
ELI5 (Fan et al., 2019)	Long-form QA	Wikipedia	ROUGE-L
FEVER (Thorne et al., 2018)	Fact checking	Wikipedia	Accuracy
WoW (Dinan et al., 2019)	Dialogue generation	Wikipedia	F1
T-REx (Elsahar et al., 2018)	Slot filling	Wikipedia	Accuracy
Out-of-domain			
zs-RE (Levy et al., 2017)	Relation extraction	Wikipedia	Accuracy
SciFact (Wadden et al., 2020)	Fact checking	BeIR	Accuracy
Climate-FEVER (Diggelmann et al., 2021)	Fact checking	BeIR	Accuracy
FiQA (Maia et al., 2018)	Financial QA	BeIR	ROUGE-L

Table 1: The datasets used in the main experiment. Climate-Fever is a four-class classification task, while the other two fact-checking tasks are binary. For metrics, NQ, HotpotQA, T-REx, and zs-RE all calculate accuracy based on exact substring matching.

4 Experimental Setup

This section introduces the main experiment setup, including datasets, baselines and implementation.

4.1 Datasets and Evaluation

We collect both in-domain and out-of-domain datasets for our experiments. In-domain datasets are utilized for providing seed data to construct synthetic training data, while out-of-domain datasets possess different tasks and corpora and are collected for further generalization tests. We collect seven datasets from KILT (Petroni et al., 2021), and three from BeIR (Thakur et al., 2021), as detailed in Table 1. All KILT datasets utilize Wikipedia dump dated 2019-08-01[2] as the corpus. Following Wang et al. (2019), we split the original articles into segments with a maximum length of 100 words, resulting in a total of 28,773,800 passages. For test sets of BeIR, we adopt their self-constructed corpora. For retrieval, we follow the closed corpus

[2] http://dl.fbaipublicfiles.com/BLINK/enwiki-pages-articles.xml.bz2

Method	In-domain						Out-of-domain			
	NQ	HotpotQA	ELI5	FEVER	WoW	T-REx	zs-RE	SciFact	C-FEVER	FiQA
LLaMA-3-8B-Instruct										
No retrieval	43.5	36.8	14.8	79.8	9.3	34.5	21.7	68.0	45.8	17.2
Contriever	43.8	36.7	14.5	78.5	8.6	33.6	20.4	70.1	38.2	16.2
BGE	47.5	41.6	15.2	**83.5**	8.7	36.4	22.7	**83.3**	44.9	21.0
$\text{AAR}_{\text{Contriever}}$	44.9	39.9	15.0	77.2	8.3	34.4	21.0	73.6	39.2	16.7
$\text{REPLUG}_{\text{Contriever}}$	43.3	38.9	13.8	80.0	9.4	33.1	22.8	74.6	41.2	18.9
$\text{SCARLet}_{\text{Contriever}}$	44.6	40.5	15.8	80.6	11.0	35.8	21.0	75.5	42.8	17.7
$\text{SCARLet}_{\text{BGE}}$	**49.2**	**47.0**	**16.3**	81.3	**12.2**	**37.0**	**24.4**	82.2	**46.1**	**22.9**
Qwen2.5-3B-Instruct										
No retrieval	27.4	26.5	**15.2**	66.1	11.5	26.0	7.3	58.2	**40.4**	17.7
Contriever	32.6	28.8	14.3	67.0	10.5	27.2	14.3	64.9	31.6	15.5
BGE	**46.8**	39.6	13.7	**78.2**	10.4	29.3	15.5	**70.6**	30.2	18.7
$\text{AAR}_{\text{Contriever}}$	34.1	29.7	13.8	66.6	10.1	28.7	15.2	63.6	32.2	16.1
$\text{REPLUG}_{\text{Contriever}}$	33.7	34.0	14.0	71.4	12.2	26.9	16.2	61.1	30.6	19.0
$\text{SCARLet}_{\text{Contriever}}$	38.2	35.4	14.9	70.8	11.7	28.0	**19.1**	65.3	31.7	17.3
$\text{SCARLet}_{\text{BGE}}$	44.9	**41.1**	**15.2**	74.3	**12.6**	**29.7**	16.6	62.3	33.0	**20.4**

Table 2: Results of the main experiment across datasets on different downstream generators. $\text{AAR}_{\text{Contriever}}$, $\text{REPLUG}_{\text{Contriever}}$, $\text{SCARLet}_{\text{Contriever}}$ represent the baselines initilized from Contriever, and $\text{SCARLet}_{\text{BGE}}$ represents the baseline initialized from BGE-base-v1.5. The **bold** score means the best performance of the corresponding dataset among baselines within the same generator, while the underline score means the second best.

setup(Asai et al., 2023), where RALMs only retrieve from the corpus of the current dataset. For the test data, we randomly sample 1,000 data from the test split of each dataset.

For evaluation metrics, we mainly assess the performance of downstream tasks. For WoW, we use F1. For ELI5 and FiQA, we use ROUGE-L. For other datasets, we use accuracy.

4.2 Baselines

The baselines are categorized into three settings:

No Retrieval The downstream generators operate without any retrieval.

Vanilla RAG Retrievers are added and the recalled passages are incorporated into the generation process. We choose two well-trained embedding models, Contriever (Izacard et al., 2022) and BGE-base-v1.5 (Xiao et al., 2023) as the retrievers.

Retriever-only Optimization Retrievers are optimized using feedback from the generator. We select two recent methods, RePlug (Shi et al., 2023) and AAR (Yu et al., 2023), both of which are initialized from Contriever.

We utilize LLaMA-3-8B-Instruct (AI@Meta, 2024) and Qwen2.5-3B-Instruct (Team, 2024) as the generators in RALMs. All retrieval-based baselines use the top-3 passages. Given that some retrievers may not be tuned by instructions, the query format for Contriever and its baselines only contains x, without task instruction I. For BGE and its baselines, the query format follows the definition in Section 3.1, which contains both x and I.

4.3 Implementation Details

In the shared context synthesis stage, we add the six tasks of the in-domain datasets into the task pool. We then randomly sample 1,000 data from the training split of each dataset to construct the seed dataset. We only consider one-hop relation when searching adjacent entities. For each entity, the top-10 passages are retrieved from $\mathcal{C}$, and the shared context is formed by selecting the top-10 passages across all retrieved passages. We utilize gpt-4o-2024-11-20 (OpenAI, 2024) as the synthesizer model. For more implementation details and meta data, please see Appendix C.

5 Results

In this section, we present the results of main experiment (§5.1), ablation study (§5.2), retrieval evaluation (§5.3), and case study (§5.4).

5.1 Overall Performance

The main experimental results are shown in Table 2. Our proposed SCARLet method achieves either optimal or suboptimal performance across various datasets and generators, demonstrating its effectiveness. Our detailed analysis from different perspectives is as follows:

In-domain Performance In the evaluation on six in-domain datasets, the retrievers trained by SCARLet achieve the best performance in five datasets when using LLaMA-3-8B as the generator, and in four datasets when using Qwen-2.5-3B as the generator. Except for NQ and FEVER, SCARLet consistently outperforms the initial baselines, including Contriever and BGE.

Out-of-domain Performance In the evaluation on four out-of-domain datasets, SCARLet also achieves optimal or suboptimal results. Specifically, SCARLet can still show progress in SciFact, Climate-FEVER, and FiQA, whose corpora differ from the Wikipedia corpus used in training and whose domains are notably different from the in-domain datasets, highlighting its generalization across corpora. In addition, SCARLet can achieve overall improvements when using two different downstream LLMs, preliminarily indicating its adaptability across generators.

5.2 Ablation Study

According to the pipeline of SCARLet, we design the ablation experiments from three stages: 1) In the data synthesis stage, we evaluate the method of removing the step of retrieving adjacent entities, and instead directly retrieving the top-k passages from the corpus $\mathcal{C}$ using only the entities extracted from the seed data; 2) In the utility attribution stage, since Section 3.4 already compares various attribution methods and demonstrates the superiority of our perturbation-based approach, we no longer conduct ablation study for this part; 3) In the sampling and training stage, we assess the effect of removing the one-dimensional clustering step and instead directly selecting the highest-scoring passage as the positive sample and the five lowest-scoring passages as negative samples based on the scores.

The comparison results, presented in Figure 4, show that removing either of the two components leads to a significant performance drop. Without adjacent entities retrieval, we believe that the original entity list may contain insufficient information, making it challenging to construct a shared context that effectively supports multi-task data synthesis. And the weaker entity associations can disrupt the connection between peer passages in the shared context, ultimately degrading the quality of the synthesized data. Furthermore, without one-dimensional clustering sampling, we suggest that it reduces the number of positive samples, which can be particularly detrimental to retrieval tasks requiring multiple reasoning hops.

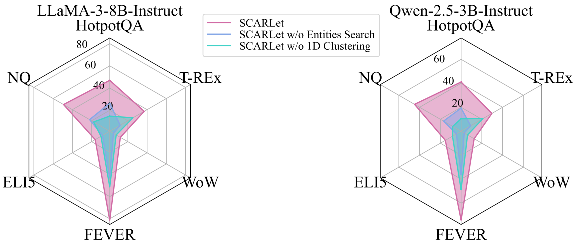


Figure 4: Ablation Study on six in-domain datasets, using BGE as retriever, with two generators. The values in the charts correspond to the metrics of each dataset.

5.3 Aspects of Retrieval Utility

The previous experiment evaluates the overall performance improvement of RALMs brought by SCARLet. However, in essence, SCARLet is an optimization method of the retrieval stage. Moreover, despite discussing the utility as the valid gain for downstream generation in RALMs, neither existing work nor this study can explicitly define utility-based retrieval. To assess the effectiveness of SCARLet in improving retrieval performance, we select three retrieval benchmarks, each representing a distinct aspect of retrieval utility based on our understanding, as shown below:

GTI This benchmark was introduced in Section 3.4. Its goal is to evaluate whether retrievers can bypass pitfalls of semantic relevance and prioritize passages that are useful for answering questions.

BRIGHT This benchmark focuses on the reasoning implied in retrieval (Su et al., 2024), particularly for complex queries that require the retriever to engage in deep reasoning to identify useful passages, beyond simple semantic relevance. Dai et al. (2024) also argue that the entailment reasoning between passages and queries is essential for enhancing retrieval capabilities. We believe that recognizing retrieval utility requires reasoning, such as distinguishing task-specific features and determining the appropriate number of hops.

$\mathbb{X}^2$-Retrieval This benchmark focuses on retrieval across multiple tasks and scenarios (Asai et al., 2023), where understanding the intent behind user's queries becomes crucial. We suggest

Method	HotpotQA		NQ		MSMARCO-QA	
	NDCG@1	NDCG@5	NDCG@1	NDCG@5	NDCG@1	NDCG@5
Contriever	33.3	48.0	10.0	35.8	16.8	37.0
SCARLet$_{\text{Contriever}}$	41.3$_{(+8.0)}$	52.1$_{(+4.1)}$	17.5$_{(+7.5)}$	45.3$_{(+9.5)}$	21.9$_{(+5.1)}$	44.1$_{(+7.1)}$
BGE	70.3	70.1	30.3	60.2	47.8	71.9
SCARLet$_{\text{BGE}}$	72.8$_{(+2.5)}$	76.7$_{(+6.6)}$	33.4$_{(+3.1)}$	64.4$_{(+4.2)}$	53.2$_{(+5.4)}$	77.0$_{(+5.1)}$

Table 3: Evaluation results on GTI, reporting nDCG for each datasets. Bracketed values indicate the changes in metrics compared to the initial model.

Model	StackExchange	Coding	Theorem-based
Contriever	10.5	19.6	6.9
SCARLet$_{\text{Contriever}}$	13.3$_{(+2.8)}$	19.2$_{(-0.4)}$	8.7$_{(+1.8)}$
BGE	14.9	16.0	8.1
SCARLet$_{\text{BGE}}$	16.2$_{(+1.3)}$	14.4$_{(-1.6)}$	9.2$_{(+1.1)}$

Table 4: Evaluation results on BRIGHT, reporting nDCG@10 for each datasets. Bracketed values indicate the changes in metrics compared to the initial model.

Model	AMB	WQA	GAT	LSO	CSP
Contriever	96.8	80.9	73.2	28.0	36.7
SCARLet$_{\text{Contriever}}$	97.5$_{(+0.7)}$	85.8$_{(+5.1)}$	71.6$_{(-1.6)}$	20.9$_{(-7.1)}$	24.8$_{(-11.9)}$
BGE	97.3	84.0	77.4	30.1	38.2
SCARLet$_{\text{BGE}}$	98.3$_{(+1.0)}$	86.1$_{(+2.1)}$	77.8$_{(+0.4)}$	27.5$_{(-2.6)}$	34.9$_{(-3.3)}$

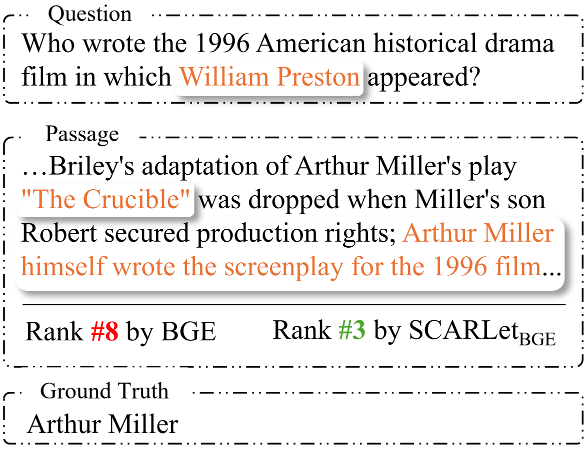
Table 5: Evaluation results on $\mathbb{X}^2$-Retrieval, averaged nDCG@10 for each datasets. Bracketed values indicate the changes in metrics compared to the initial model.

that this corresponds to identifying the target utility anticipated by the downstream tasks.

We choose Contriever and BGE as the retriever models, using LLaMA-3-8B-Instruct as the downstream generator to implement SCARLet training. We compare the performance of the trained retrievers with the initial retrievers on two benchmarks, as shown in Table 3, 4 and 5, respectively. The results indicate that SCARLet improves performance on some datasets, but its effectiveness is generally limited for code-related tasks, such as LinkSo (Liu et al., 2018) and CodeSearchNet (Husain et al., 2020). The reasons could be: 1) the significant difference between the code domain and our selected in-domain datasets, which may hinder generalization; 2) the retriever models used are relatively lightweight, making it susceptible to catastrophic forgetting during training; 3) the optimization is related to downstream generators, but feedback related to the code domain cannot be obtained.

5.4 Case Study

Multi-hop QA is a task that requires multiple pieces of information and multi-step reasoning to derive the solution (Mavi et al., 2024). Given the characteristics of the task, we believe that retrieval utility should point to passages that may contain information necessary for the reasoning chain. We select a representative example from the test split of the HotpotQA dataset, as shown in Figure 5. To answer the question, the reasoning chain is: knowing information about William Preston, identifying the 1996 American historical drama he appeared in, finding information about that drama, and determining its writer. Directly relevant information about William Preston is relatively easy to define. However, the shown passage which corresponds to the final reasoning step, has a poor match with the question in terms of semantic relevance. And BGE ranks it 8th. After training by SCARLet, the passage achieves a higher ranking of 3rd. For more case studies, please refer to Appendix E.


Question

Who wrote the 1996 American historical drama film in which William Preston appeared?

Passage

…Briley's adaptation of Arthur Miller's play "The Crucible" was dropped when Miller's son Robert secured production rights; Arthur Miller himself wrote the screenplay for the 1996 film...

Rank #8 by BGE   Rank #3 by SCARLet$_{\text{BGE}}$

Ground Truth

Arthur Miller

Figure 5: Case Study on HotpotQA. The passage is ranked variously by different retrievers. Orange text indicates necessary reasoning information.

6 Conclusion

This study focuses on utility-based retrieval, a paradigm that moves beyond semantic relevance to prioritize downstream task performance in RALMs. We highlight two key challenges faced by existing research. To solve the limitations, we propose SCARLet, a novel framework to enhance utility-based retrieval. To mitigate semantic interference on utility features during training, SCARLet incorporates a shared context synthesis method, which narrows the semantic gap between different tasks. To address the issue of inaccurate passage-level utility estimation, SCARLet employs a perturbation-based attribution method to capture the synergy between passages. Lastly, SCARLet utilizes a one-dimensional clustering method to sample positive and negative passages for retriever optimization. Through experiments, we demonstrate that SCARLet can effectively enhance the overall performance of RALMs, and brings improvements in complex retrieval benchmarks. We hope this study can inspire further research on utility-based retrieval.

Limitations

This study only covers several classic downstream datasets. We believe that incorporating a task augmentation stage could further enhance generalization, which we leave for future work. Moreover, there is a noticeable decline in retrieval performance in the code domain during generalization tests. Therefore, future work should also focus on improving the integration of different corpus structures. In addition, due to environmental constraints, this study does not evaluate larger-scale retrievers and generators. Furthermore, we also try GPT-4o-mini as the synthesizer, which performed poorly. Thus our framework should be equipped with models with stronger reasoning capabilities.

Ethics Statement

All datasets and corpora involved in this study are publicly available, and we ensure that all used data comply with the usage and privacy policies established by the original authors. The synthetic data is exclusively used for training the retriever model. Moreover, given the security assurance of the synthesizer model, the probability of generating harmful passages and data is extremely minimal.

Acknowledgments

This work was supported by the National Key R&D Program of China (2023YFC3303800).

References

AI@Meta. 2024. Llama 3 model card.

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2023. Task-aware retrieval with instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3650–3675, Toronto, Canada. Association for Computational Linguistics.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Hongcheng Gao, Yilong Xu, and Xueqi Cheng. 2024. Adaptive token biaser: Knowledge editing via biasing key entities. *Preprint*, arXiv:2406.12468.

Baolong Bi, Shenghua Liu, Yiwei Wang, Yilong Xu, Junfeng Fang, Lingrui Mei, and Xueqi Cheng. 2025. Parameters vs. context: Fine-grained control of knowledge reliance in language models. *arXiv preprint arXiv:2503.15888*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Canyu Chen and Kai Shu. 2024. Can llm-generated misinformation be detected? *Preprint*, arXiv:2309.13788.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self-memory. In *Advances in Neural Information Processing Systems*, volume 36, pages 43780–43799. Curran Associates, Inc.

Lu Dai, Hao Liu, and Hui Xiong. 2024. Improve dense passage retrieval with entailment tuning. *Preprint*, arXiv:2410.15801.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2015. Extraction of salient sentences from labelled documents. *Preprint*, arXiv:1412.6815.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. Climate-fever: A dataset for verification of real-world climate claims. *Preprint*, arXiv:2012.00614.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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, Florence, Italy. Association for Computational Linguistics.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. *Preprint*, arXiv:2405.20978.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *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, Seattle, United States. Association for Computational Linguistics.

Jiafeng Guo, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2019. Matchzoo: A learning, practicing, and developing system for neural text matching. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1297–1300, New York, NY, USA. Association for Computing Machinery.

Xiaochuang Han and Yulia Tsvetkov. 2022. Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. *Preprint*, arXiv:2205.12600.

Donald E. Hilt and Donald W. Seegrist. 1977. Ridge: a computer program for calculating ridge regression estimates.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2020. Codesearchnet challenge: Evaluating the state of semantic code search. *Preprint*, arXiv:1909.09436.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Minsang Kim and Seungjun Baek. 2025. Syntriever: How to train your retriever with synthetic data from llms. *Preprint*, arXiv:2502.03824.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Liu, Ziyang Chen, Baotian Hu, Aiguo Wu, and Min Zhang. 2023. A survey of large language models attribution. *Preprint*, arXiv:2311.03731.

Xinze Li, Yixin Cao, Liangming Pan, Yubo Ma, and Aixin Sun. 2024. Towards verifiable generation: A benchmark for knowledge-aware language model attribution. *Preprint*, arXiv:2310.05634.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. 2024. Radit: Retrieval-augmented dual instruction tuning. *Preprint*, arXiv:2310.01352.

Xueqing Liu, Chi Wang, Yue Leng, and ChengXiang Zhai. 2018. Linkso: a dataset for learning to retrieve similar question answer pairs on software development forums. In *Proceedings of the 4th ACM SIGSOFT International Workshop on NLP for Software Engineering*, NL4SE 2018, page 2–5, New York, NY, USA. Association for Computing Machinery.

Yuhang Liu, Xueyu Hu, Shengyu Zhang, Jingyuan Chen, Fan Wu, and Fei Wu. 2024. Fine-grained guidance for retrievers: Leveraging llms' feedback in retrieval-augmented generation. *Preprint*, arXiv:2411.03957.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On llms-driven synthetic data generation, curation, and evaluation: A survey. *Preprint*, arXiv:2406.15126.

Gianluigi Lopardo, Frederic Precioso, and Damien Garreau. 2024. Attention meets post-hoc interpretability: A mathematical perspective. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32781–32800. PMLR.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 295–337, Bangkok, Thailand. Association for Computational Linguistics.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 1941–1942, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Dina Mardaoui and Damien Garreau. 2021. An analysis of lime for text data. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3493–3501. PMLR.

Vaibhav Mavi, Anubhav Jangra, and Adam Jatowt. 2024. Multi-hop question answering. *Preprint*, arXiv:2204.09140.

Nikolaos Mylonas, Ioannis Mollas, and Grigorios Tsoumakas. 2022. An attention matrix for every decision: Faithfulness-based arbitration among multiple attention-based interpretations of transformers in text classification. *Preprint*, arXiv:2209.10876.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models. *Preprint*, arXiv:2307.06435.

Ian E. Nielsen, Dimah Dera, Ghulam Rasool, Ravi P. Ramachandran, and Nidhal Carla Bouaynaya. 2022. Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *IEEE Signal Processing Magazine*, 39(4):73–84.

OpenAI. 2024. Gpt-4o system card.

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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-agnostic interpretability of machine learning. *Preprint*, arXiv:1606.05386.

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

Alireza Salemi and Hamed Zamani. 2024. Towards a search engine for machines: Unified ranking for multiple retrieval-augmented large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 741–751, New York, NY, USA. Association for Computing Machinery.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *Preprint*, arXiv:2305.15294.

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. *Preprint*, arXiv:2301.12652.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiwoong Sohn, Yein Park, Chanwoong Yoon, Sihyeon Park, Hyeon Hwang, Mujeen Sung, Hyunjae Kim, and Jaewoo Kang. 2024. Rationale-guided retrieval augmented generation for medical question answering. *Preprint*, arXiv:2411.00300.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O. Arik, Danqi Chen, and Tao Yu. 2024. Bright: A realistic and challenging benchmark for reasoning-intensive retrieval. *Preprint*, arXiv:2407.12883.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. 2024. Gradient based feature attribution in explainable ai: A technical review. *Preprint*, arXiv:2403.10415.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. InstructRAG: Instructing retrieval augmented generation via self-synthesized rationales. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.

Orion Weller, Marc Marone, Nathaniel Weir, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. “according to . . . ”: Prompting language models improves quoting from pre-training data. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2288–2301, St. Julian’s, Malta. Association for Computational Linguistics.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. 2024. How easily do irrelevant inputs skew the responses of large language models? *Preprint*, arXiv:2404.03302.

Long Xia, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. 2015. Learning maximal marginal relevance model via directly optimizing diversity evaluation measures. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’15, page 113–122, New York, NY, USA. Association for Computing Machinery.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *Preprint*, arXiv:2007.00808.

Yilong Xu, Jinhua Gao, Xiaoming Yu, Baolong Bi, Huawei Shen, and Xueqi Cheng. 2024. Aliice: Evaluating positional fine-grained citation generation. *Preprint*, arXiv:2406.13375.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

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. *Preprint*, arXiv:2407.02485.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2421–2436, Toronto, Canada. Association for Computational Linguistics.

Hamed Zamani and Michael Bendersky. 2024. Stochastic rag: End-to-end retrieval-augmented generation through expected utility maximization. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’24, page 2641–2646, New York, NY, USA. Association for Computing Machinery.

Hengran Zhang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2024. Are large language models good at utility judgments? In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’24, page 1941–1951, New York, NY, USA. Association for Computing Machinery.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Details of Data Synthesis

The detailed steps of the data synthesis pipeline in SCARLet are described as follows:

Seed Datasets Collection We first collect seed data for the data synthesis pipeline. A task pool is defined, including the selected tasks and their corresponding datasets. Each task is associated with task instruction and retrieval instruction, as shown in Table 6. For each dataset, we randomly sample 1,000 instances from its training split, including both input and ground truth. Every sampling uses the same random seed for every dataset.

Entities Extraction For each seed data instance, we extract entities for subsequent passages retrieval. We utilize the SpaCy[3] toolkit to extract entities from both the input and ground truth. Data instances without extractable entities are discarded.

Entities Retrieval This stage is to retrieve more relevant entities based on the extracted ones. This serves two purposes: 1) to enhance diversity, and 2) to strengthen relationships between entities, facilitating better construction of the shared context. We retrieve neighboring entities from Wikidata, considering only the direct related entities of each existing entity. To achieve this, we write the SPARQL query for retrieval, as shown below:

```
SELECT ?property ?propertyLabel ?object
    ?objectLabel
WHERE {
    wd:{id} ?property ?object.
    ?property rdfs:label ?propertyLabel.
    ?object rdfs:label ?objectLabel.
    FILTER(LANG(?propertyLabel) = "en")
    FILTER(LANG(?objectLabel) = "en")
}
LIMIT {limit}
```

Passages Retrieval After obtaining the expanded entity list, we retrieve relevant passages based on these entities to construct the shared context.

Data Synthesis At this stage, training data is synthesized for different tasks in the task pool based on the shared context. First, a synthesizer model is selected, which must possess sufficient reasoning and generation capabilities to ensure the quality of the synthetic data. To help the synthesizer understand the task definition and follow the correct format, we provide task instruction, task description, and example data in the prompt. The synthetic data should include both the input and ground truth. The prompt template we use is shown in Table 9.

Data Filtering In this stage, the data synthesized in the previous phase is cleaned to further ensure data quality and training stability. We prompt the synthesizer model to check the synthetic data for logical consistency and format correctness based on the shared context. The prompt used for this stage is shown in Table 10.

Passages Enhancement To enhance the robustness of the training, we inject noise into the shared context. We instruct the synthesizer model to generate a passage that is semantically relevant but useless for downstream task, and then add this passage to the shared context. The prompt used for this stage is shown in Table 11.

B Details of Utility Attribution

Introduction to Attribution Attribution is a local-interpretable technique used to provide evidence for the model generation (Li et al., 2023; Xu et al., 2024). The data source of attribution can be training data (Han and Tsvetkov, 2022; Weller et al., 2024), whereas in RALMs, the source is often retrieved external passages (Shuster et al., 2021; Li et al., 2024), which we denote as context attribution. Furthermore, contributive attribution is a form of attribution that quantifies the contribution of each data source unit to the generation process. It assigns an attribution score to each unit, where a higher score indicates a greater contribution. In this study, we propose the SCARLet framework, which employs a perturbation-based attribution method to estimate the utility score of each passage within the shared context. Additionally, we evaluate other attribution methods, including attention-based method, gradient-based method, and LLM-based method.

Perturbation-based Method This method is described in Section 3.4. Notably, unlike the classical LIME method, we remove the weight of $\mathbf{v}_i$ in the surrogate model, which measures the the cosine distance from the original text. The reason behind this is that for different perturbation vectors, the weight would exacerbate the unfair evaluation of passage utility, as utility features cannot be directly measured by semantic relevance. For passages that are semantically relevant but essentially useless, the variation they bring would be downweighted,

[3] https://spacy.io/

Dataset	Task	Task Instruction	Retrieval Instruction
NQ	Single-hop QA	`Answer the question based on the given passages.`	`Retrieve passages to answer the question.`
HotpotQA	Multi-hop QA	`Answer the question based on the given passages. You may need to refer to multiple passages.`	`Find passages that provide useful information to answer this question.`
ELI5	Long-form QA	`Answer the question based on the given passages. The answer needs to be detailed, paragraph-level, and with explanations.`	`Retrieve passages that provide a piece of good evidence for the answer.`
FEVER	Fact Checking	`Verify whether the claim is correct based on the given passages. If it is correct, output "SUPPORTS", if it is wrong, output "REFUTES".`	`Retrieve passages to verify this claim.`
WoW	Dialogue Generation	`Generate an appropriate, reasonable and meaningful response based on previous conversations and the following relevant passages.`	`Find passages related to the conversation topic.`
T-REx	Slot Filling	`Given an entity and an attribute (or relationship), fill in the specific value of the attribute based on the following passages. The entity and the attribute are separated by "[SEP]".`	`Find passages related to the entities.`
SciFact	Fact Checking	`Verify whether the claim is correct based on the given passages. If it is correct, output "SUPPORT", if it is wrong, output "CONTRADICT".`	`Retrieve passages to verify this claim.`
zs-RE	Relation Extraction	`Given an entity and an attribute (or relationship), fill in the specific value of the attribute based on the following passages. The entity and the attribute are separated by "[SEP]".`	`Find passages related to the entities.`
FiQA	Financial QA	`Answer the question based on the given passages.`	`Find passages to answer the question.`
Climate-FEVER	Fact Checking	`Verify whether the claim is correct based on the given passages. If it is correct, output "SUPPORTS", if it is wrong, output "REFUTES", if the information is insufficient, output "NOT_ENOUGH_INFO", if can't get a sufficiently confident judgment, output "DISPUTED".`	`Retrieve passages to verify this claim.`

Table 6: Task instructions and retrieval instructions of the datasets in the task pool.

```
Please first provide the answer based on the
passages that you have ranked in utility and then
write the ranked passages in descending order of
utility in answering the question, like "My rank:
[i]>[j]>...>[k]".

Context: {context}


Question: {query}
```

Table 7: The prompt template for LLM-based method.

Method	HotpotQA		NQ		MSMARCO-QA	
	NDCG@1	NDCG@5	NDCG@1	NDCG@5	NDCG@1	NDCG@5
Att.-based	31.54	27.25	29.14	25.77	29.92	22.15
Grad.-based	49.90	38.83	50.58	44.56	59.09	53.35
LLM-based	76.34	76.84	28.35	32.16	31.88	59.97
Pert.-based	**93.28**	**83.04**	**78.16**	**84.12**	**91.65**	**85.36**
w/o G.T.	92.34	81.03	77.85	80.67	91.10	83.73

Table 8: The experimental results comparing various utility attribution methods on the GTI benchmark. Attn., Grad., Pert., and G.T. represent Attention, Gradient, Perturbation and Ground Truth, respectively.

as such passages typically cause greater logit fluctuations due to their lack of utility.

Attention-based Method This method takes the attention score received by each source unit during inference as the attribution score (Mylonas et al., 2022; Lopardo et al., 2024). We construct the attention-based baseline by averaging the attention values of each token within each passage, as shown below:

$$\boldsymbol{\alpha}_{d_i} = \frac{1}{K \cdot |d_i|} \sum_{t \in d_i} \sum_{i=1}^{K} a_t^{(i)}, t \in d_i, \qquad (7)$$

where $\boldsymbol{\alpha}_{d_i}$ represents the utility score for passage d_i, K indicates the number of attention heads, and $a_t^{(i)}$ indicates the attention value of the t-th token in passage d_i of the i-th attention head.

Gradient-based Method This approach determines the utility scores from the gradient of each token in the source unit during backward propagation(Nielsen et al., 2022; Wang et al., 2024). Specifically, we employ the Gradient times Input ($G \times I$; Denil et al., 2015), which computes the score of each token by performing the dot product as follows:

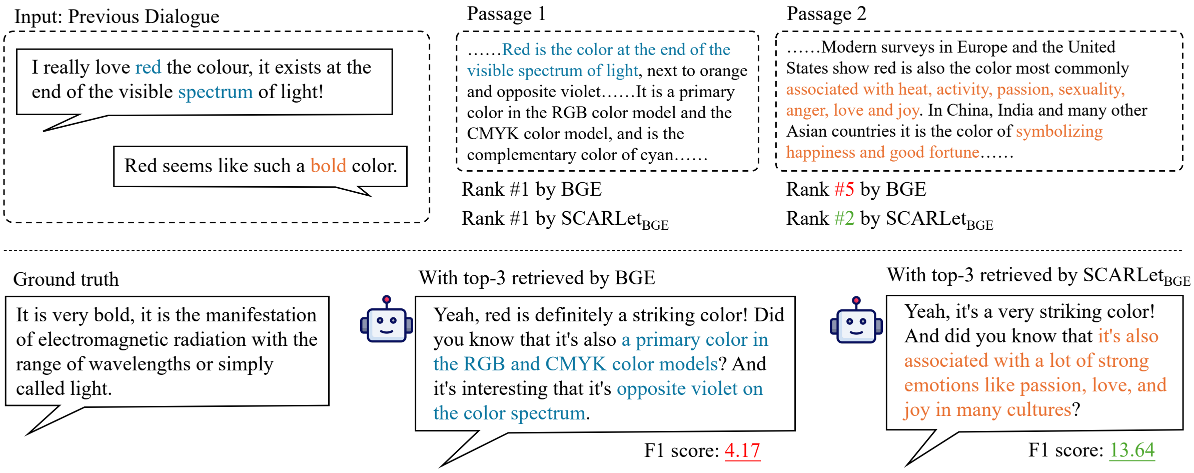
$$f_{G \times I}(t) = e_t \cdot \nabla_{e_t} f_{\text{LM}}(x, D), \qquad (8)$$

where e_t represents the embedding vector of token t, and f_{LM} denotes the function of LM. The utility score of each passage is then obtained by averaging the $G \times I$ scores of each token contained within it.

LLM-based Method This approach, which can also be referred to as rationale-based method or self-rationalization, is in line with the work of Sohn et al. (2024); Wei et al. (2024), where the LLM generator simultaneously attributes the utility of passages in the context while performing the task. Although this method is theoretically flawed due to the potential influence of hallucinations from LLMs (Chen and Shu, 2024), we still believe that it represents one of the future directions of utility attribution. Following Zhang et al. (2024), we instruct the generator to rank the passages from the context in a list-wise setup while generating the answer. The prompt is shown at Table 7.

GTI Benchmark This benchmark (Ground-Truth Inclusion; Zhang et al., 2024) is designed to assess the utility of retrieved passages including three QA datasets: NQ, with 1,868 data; HotpotQA, with 4,407 data; and MSMARCO-QA, with 3,121 data. It manually constructs 10 passages per query, including ground truth (correct passages), counterfactual passages, highly relevant noisy passages, and weakly relevant noisy passages. We evaluate the above methods on this benchmark using LLaMA-3-8B-Instruct as the generator, with the experimental results presented in Table 8. The results demonstrate that the perturbation-based method outperforms all other baselines by a significant margin, highlighting its considerable advantage as an indicator for utility in RALMs.

Attribution Forms Additionally, we investigate two different attribution forms: 1) The first form directly uses the ground truth provided by the dataset as the output of the generator, which is adopted in our proposed SCARLet; 2) The second form is let the generator to produce a response first, followed by attribution based on that response. The first form reflects the contribution of each passage within the context to the production of the correct answer. While the second form requires an additional comparison between the generated response and the ground truth, where we believe that the attribution process can be valid only if the the two are consistent. We compare the performance of the above two forms in the perturbation-based method, as shown in Table 8. We find that the performance difference between the two forms is minimal, but in terms of mechanism and difficulty of implemen-



Input: Previous Dialogue

I really love red the colour, it exists at the end of the visible spectrum of light!

Red seems like such a bold color.

Passage 1

……Red is the color at the end of the visible spectrum of light, next to orange and opposite violet……It is a primary color in the RGB color model and the CMYK color model, and is the complementary color of cyan……

Rank #1 by BGE

Rank #1 by $\text{SCARLet}_{\text{BGE}}$

Passage 2

……Modern surveys in Europe and the United States show red is also the color most commonly associated with heat, activity, passion, sexuality, anger, love and joy. In China, India and many other Asian countries it is the color of symbolizing happiness and good fortune……

Rank #5 by BGE

Rank #2 by $\text{SCARLet}_{\text{BGE}}$

Ground truth

It is very bold, it is the manifestation of electromagnetic radiation with the range of wavelengths or simply called light.

With top-3 retrieved by BGE

Yeah, red is definitely a striking color! Did you know that it's also a primary color in the RGB and CMYK color models? And it's interesting that it's opposite violet on the color spectrum.

F1 score: 4.17

With top-3 retrieved by $\text{SCARLet}_{\text{BGE}}$

Yeah, it's a very striking color! And did you know that it's also associated with a lot of strong emotions like passion, love, and joy in many cultures?

F1 score: 13.64

Figure 6: Case Study on WoW. Blue text indicates clues more relevant to semantics, while orange text highlights clues more align with the target utility in dialogue generation task. Responses are generated by LLaMA-3-8B. The generated response augmented by $\text{SCARLet}_{\text{BGE}}$ achieves a higher F1 score than the response augmented by BGE.

tation, we choose the first form.

C Details of Implementation

Meta data of data synthesis We present the meta data from the data synthesis pipeline of one run in our experiment, as shown in Table 12. As observed, although the amount of training data is sufficient for tuning the retriever, the SCARLet pipeline leads to data loss at each stage, sometimes resulting in significant loss rates, which causes an increase in costs. The reasons for the loss include issues with the seed data, network problems, model generation errors, among others.

Hyperparameters During the data synthesis stage, the temperature of the synthesizer model is set to 0.5. In the utility attribution stage, the number of sampled perturbation vectors n is set to 64, with a perturbation probability of 0.5. During training, we set the learning rate as 6e-5, and epochs as 1. All experiments are conducted on NVIDIA A100 GPUs in torch.float32 precision.

D Additional Experimental Results

The results presented in Table 2 are under the closed corpus setup, i.e., retrievers search passages only from the corpus of the corresponding dataset. In contrast, the pooled corpus setup refers to merging the corpora of different datasets into a single corpus, where all retrieval is performed with the unified corpus. This setup better simulates real-world retrieval scenarios and enables a fairer evaluation of generalization. The experimental results under the pooled corpus setup are shown in Table 13. All baselines perform similarly to those in the closed corpus setup, and some outperform them, demonstrating generalization of SCARLet on the unified corpus.

E Additional Case Study

The QA tasks typically focus more on precise answers, whereas dialogue tasks prioritize the coherence between the generated response and the preceding conversation. These two tasks have distinct retrieval utility, with the latter being more vaguely defined. To analyze whether the retriever trained by SCARLet exhibits a diversified retrieval criteria, we select a case from the test split of the WoW dataset, as shown in Figure 6. In this case, a retriever relying on semantic relevance may primarily focus on topic words such as "red" and "spectrum". However, for dialogue generation, it is also crucial to consider the intent of the previous speaker. Passage 2 is ranked higher by the retriever trained by SCARLet, because it is directly tied to the deeper meaning of the key clue "bold", making it more helpful in sustaining conversational coherence. At comparable recall levels, SCARLet prioritizes passages that offer greater task-specific utility.

F Example of Shared Context

In this section, we provide an example of the shared context constructed during one run of SCARLet in our experiment, as shown in Figure 7, along with its corresponding synthetic data for various tasks, as shown in Figure 8 and Figure 9.

Shared Context Example

[1]: Deutsche Telekom AG (; short form in writing only: DT; stylised as ·T···) is a German telecommunications company headquartered in Bonn and by revenue the largest telecommunications provider in Europe. Deutsche Telekom was formed in 1996, as the former state-owned monopoly Deutsche Bundespost was privatised. The company operates several subsidiaries worldwide, including the mobile communications brand T-Mobile. As of 2020, the German government holds a 14.5% stake in company stock directly, and another 17.4% through the government bank KfW. The company is a component of the Euro Stoxx 50 stock market index.

[2]: AT&T was broken up in 1984. In the case of Telecom New Zealand, local loop unbundling was enforced by central government. Telkom is a semi-privatised, part state-owned South African telecommunications company. Deutsche Telekom is a former state monopoly, still partially state owned. Deutsche Telekom currently monopolizes high-speed VDSL broadband network. The Long Island Power Authority (LIPA) provided electric service to over 1.1 million customers in Nassau and Suffolk counties of New York, and the Rockaway Peninsula in Queens. The Comcast Corporation is the largest mass media and communications company in the world by revenue.

[3]: Telkom Indonesia PT Telekomunikasi Indonesia (Persero) Tbk, commonly known as Telkom Indonesia (stylised as Telkom Indonesıa) or simply Telkom, is an Indonesian multinational telecommunications conglomerate. Telkom is a semi-privatised, majority state-owned company listed on multiple exchanges. It has major businesses in fixed line telephony, internet and data communications. It is operated as parent company of the Telkom Group, which is engaged in a broad range of businesses that consist of telecommunication, multimedia, property and finance services. Since 2008, Telkom Indonesia began changing its business focus, infrastructure, systems, organisation and human resources, as well as the corporate culture, as their effort

[4]: Mannesmann Arcor AG & Co. KG was formed in 1996 as a joint venture between Mannesmann, Deutsche Bank and DBKom, a subsidiary of Deutsche Bahn, the national railway operator. After Mannesmann was bought out by Vodafone the company was renamed Arcor AG & Co. KG. On 19 May 2008, Vodafone acquired the minority shareholdings of Deutsche Bahn and Deutsche Bank (18.17% and 8.18% respectively) to gain full control of Arcor.

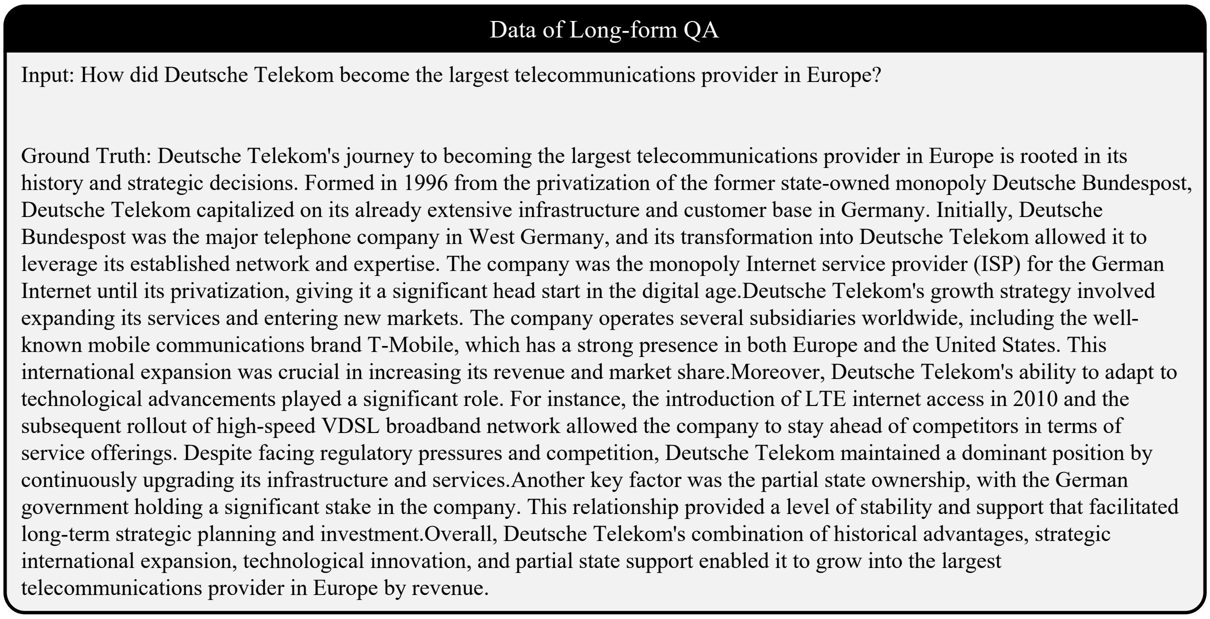
[5]: Up to 1998, the telecommunications market in Greece was a monopoly. The market was opened to competitors and OTE was gradually privatized. As a result, the Greek government currently holds 10% of the company's stock. Along with the rest of the telecoms market, OTE is regulated by the National Telecommunications and Post Commission (Εθνική Επιτροπή Τηλεπικοινωνιών και Ταχυδρομείων – EETT). In 2007 Marfin Investment Group acquired 20% of the company, and in March 2008 sold it to Germany's Deutsche Telekom which later increased its stake to 25% plus

[6]: federal German government postal administration created in 1947 as a successor to the Reichspost. It was also the major telephone company in West Germany. On 1 July 1989, as part of a post office reform, Deutsche Bundespost was split into three entities, one being Deutsche Telekom. On 1 January 1995, as part of another reform, Deutsche Bundespost Telekom became Deutsche Telekom AG, and was privatized in 1996. As such, it shares a common heritage with the other privatized Deutsche Bundespost companies, Deutsche Post (DHL) and Deutsche Postbank. Deutsche Telekom was the monopoly Internet service provider (ISP) for the German Internet

[7]: This was important because local telephone calls in Germany, including dialup access to ISPs, were not offered on a flat price per call (i.e., unlimited) basis. The flat-rate service was also offered to customers with ISDN connections at the same price as for analog service. In Spring 2001, T-Online announced the demise of the flat-rate dialup plan but offered a flat-rate DSL plan in its place. Deutsche Telekom (T-Online) was the monopoly Internet Service Provider (ISP) for the German Internet until its privatization in 1995, and the dominant ISP thereafter.

[8]: Telefónica Germany Telefónica Germany GmbH & Co. OHG (; also called Telefónica Deutschland) is a provider of broadband, landline and mobile telecommunications in Germany. The company trades as O2 (typeset as O) and O2 DSL (formerly Alice). The company was renamed from Telefónica O2 Germany to Telefónica Germany on 1 April 2011 following the completion of a merger with HanseNet. Telefónica Germany purchased E-Plus on 1 October 2014, unifying the business under O brand on 3 February 2016. Telefónica Germany's main competitors are Telekom Deutschland (Deutsche Telekom's German private customer unit) and Vodafone.

Figure 7: An example of the shared context. Based on this context, SCARLet synthesizes training data, as shown in Figure 8 and 9.

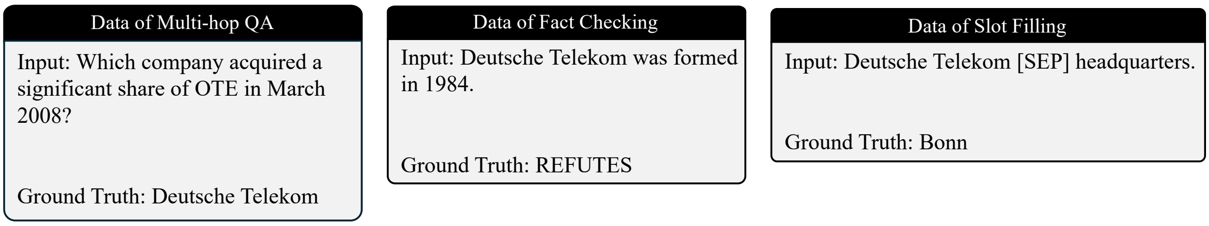


Data of Long-form QA

Input: How did Deutsche Telekom become the largest telecommunications provider in Europe?

Ground Truth: Deutsche Telekom's journey to becoming the largest telecommunications provider in Europe is rooted in its history and strategic decisions. Formed in 1996 from the privatization of the former state-owned monopoly Deutsche Bundespost, Deutsche Telekom capitalized on its already extensive infrastructure and customer base in Germany. Initially, Deutsche Bundespost was the major telephone company in West Germany, and its transformation into Deutsche Telekom allowed it to leverage its established network and expertise. The company was the monopoly Internet service provider (ISP) for the German Internet until its privatization, giving it a significant head start in the digital age.Deutsche Telekom's growth strategy involved expanding its services and entering new markets. The company operates several subsidiaries worldwide, including the well-known mobile communications brand T-Mobile, which has a strong presence in both Europe and the United States. This international expansion was crucial in increasing its revenue and market share.Moreover, Deutsche Telekom's ability to adapt to technological advancements played a significant role. For instance, the introduction of LTE internet access in 2010 and the subsequent rollout of high-speed VDSL broadband network allowed the company to stay ahead of competitors in terms of service offerings. Despite facing regulatory pressures and competition, Deutsche Telekom maintained a dominant position by continuously upgrading its infrastructure and services.Another key factor was the partial state ownership, with the German government holding a significant stake in the company. This relationship provided a level of stability and support that facilitated long-term strategic planning and investment.Overall, Deutsche Telekom's combination of historical advantages, strategic international expansion, technological innovation, and partial state support enabled it to grow into the largest telecommunications provider in Europe by revenue.

Figure 8: The training data of long-form QA, synthesized by SCARLet based on the context in Figure 7.



Data of Multi-hop QA

Input: Which company acquired a significant share of OTE in March 2008?

Ground Truth: Deutsche Telekom

Data of Fact Checking

Input: Deutsche Telekom was formed in 1984.

Ground Truth: REFUTES

Data of Slot Filling

Input: Deutsche Telekom [SEP] headquarters.

Ground Truth: Bonn

Figure 9: The training data of multi-hop QA, fact checking and slot filling, synthesized by SCARLet based on the context in Figure 7.

```
You are a strong expert of data synthesis. Below, I will provide the context, the
description and an example of the target task. Your task is to generate a piece of data for
the target task based on the given context. The sections marked with ====xxx begins==== and
====xxx ends==== indicate the start and end of each respective part. Please note that the
data you generate must meet the following criteria:
1. Correctness, which must be logically correct and factually correct.
2. Faithfulness, which must be faithful to the context.
3. Quality, which must be thoughtful and sophisticated, ideally based on multiple
paragraphs where applicable.

Please note that the generated data should follow this specific format:
====New data begins====
Input:
Reference output:
====New data ends====

====Context begins====
{context}
====Context ends====

====Target task description begins====
{task_description}
====Target task description ends====

====Target task example begins====
Input: {task_example_input}
Reference output: {task_example_output}
====Target task example ends====

Please ensure that your output matches the instructions above.
```

Table 9: The prompt template for data synthesis.

```
You are tasked with checking whether the following synthetic data of {task_name} task is
logically correct and formatted correctly. The data consists of five parts: task
description, example, input, output, source passages. The input and output of the synthetic
data are based on the source passages. And a reasonable example of {task_name} task is
provided, note that it is not based on source passages. Please check the following:
1. Logical Correctness: Check whether the output correctly solves the input based on the
source passages.
2. Format Correctness: Check whether the input and output of the synthetic data conform to
the correct format presented in the task description and the example.

Task description: {task_description}

Example:
Input: {task_example_input}
Output: {task_example_output}

Now, please check the following synthetic data based on source passages:

Input: {input}
Output: {output}
Source passages: {context}

Please note that if the above synthetic data basically meets the requirements, output
"[YES]", otherwise output "[NO]".
```

Table 10: The prompt template for data filtering.

```
You are a strong expert of data processing. You are tasked with data augmentation to
generate noisy data to enhance training robustness. Below, I will provide you with a piece
of data, including task description, input, and ground truth. Then I will provide you with
the context containing the necessary information to solve the input. You need to deeply
understand the data and the context, and finally generate a passage which is a variant of
one passage of the context. The generated passage needs to be semantically relevant while
providing no practical effect in solving the input.

Data:
Input: {data_input}
Ground truth: {data_output}

Context: {context}

Please ensure that the generated passage matches the length of the passages in the context
and is a modified version of its original passage. And the generated passage must follow
the format, which is marked with ====Generated passage begins==== and ====Generated passage
ends==== at its start and end.
```

Table 11: The prompt template for passages enhancement.

	NQ	HotpotQA	ELI5	FEVER	WoW	T-REx
Entities Extraction						
Loss Rate	12.2%	0.6%	24.1%	5.1%	17.1%	9.2%
Averaged Number of Entities	1.7	3.4	5.8	2.0	5.1	1.8
Entities Retrieval						
Expansion Rate	91.0%	90.5%	96.0%	89.1%	97.5%	98.0%
Averaged Number of New Entities	5.1	6.5	17.1	3.7	14.9	5.2
Averaged Number of Entities	6.3	9.3	22.2	5.3	19.6	6.9
Data Synthesis						
Number of Synthetic Data	5230	5950	4492	5580	4872	5317
Loss Rate	12.8%	0.8%	25.1%	7.0%	18.8%	11.4%

Table 12: Meta data from the synthesis pipeline of one run in our experiment. Loss Rate means the proportion of discarded data caused by the process. Expansion Rate means the proportion of data with new entities added. In this run, the data filtering achieves a loss rate of 44.2%, and the total amount of data used for utility attribution is 17,529.

Method	**In-domain**						**Out-of-domain**			
	NQ	**HotpotQA**	**ELI5**	**FEVER**	**WoW**	**T-REx**	**zs-RE**	**SciFact**	**C-FEVER**	**FiQA**
LLaMA-3-8B-Instruct										
Contriever	44.0	36.7	14.5	79.2	8.6	33.8	20.9	68.1	38.0	16.5
BGE	48.0	45.4	15.2	**85.6**	8.8	**39.6**	24.1	80.2	**45.9**	20.8
AAR$_{\text{Contriever}}$	46.2	41.8	15.0	77.8	8.2	35.1	24.2	70.3	42.6	16.7
REPLUG$_{\text{Contriever}}$	44.5	39.7	13.8	81.3	9.2	33.7	23.6	72.9	41.0	18.8
SCARLet$_{\text{Contriever}}$	45.1	42.0	15.9	80.6	10.4	36.4	22.2	74.7	42.0	17.7
SCARLet$_{\text{BGE}}$	**49.8**	**48.3**	**16.6**	81.2	**12.7**	37.0	**24.7**	**81.5**	**45.9**	**23.1**
Qwen2.5-3B-Instruct										
Contriever	31.9	28.5	14.2	67.1	10.5	27.1	14.0	**66.5**	32.8	15.5
BGE	**48.5**	44.0	13.7	**80.4**	10.2	**34.5**	18.6	65.5	**37.1**	18.6
AAR$_{\text{Contriever}}$	34.8	30.9	13.8	66.2	10.6	28.3	15.5	63.2	32.0	16.3
REPLUG$_{\text{Contriever}}$	34.2	35.8	14.0	71.2	**12.8**	26.8	16.9	60.6	30.9	18.7
SCARLet$_{\text{Contriever}}$	39.3	36.0	14.4	70.0	11.9	28.2	**19.1**	64.9	31.8	17.3
SCARLet$_{\text{BGE}}$	45.1	**44.7**	**15.6**	74.1	12.3	30.1	18.7	64.4	36.3	**20.5**

Table 13: Results of the main experiment in the pooled corpus setup. The unified corpus includes corpora of Wikipedia dump, BeIR-SciFact, BeIR-ClimateFEVER and BeIR-FiQA.