GeAR: Generation Augmented Retrieval

Anonymous ACL submission

Abstract

Document retrieval techniques are essential for developing large-scale information systems. The common approach involves using a bi-encoder to compute the semantic similarity between a query and documents. However, the scalar similarity often fail to reflect enough information, hindering the interpretation of retrieval results. In addition, this process primarily focuses on global semantics, overlooking the finer-grained semantic relationships between the query and the document's content. In this paper, we introduce a novel method, **Ge**neration **A**ugmented **R**etrieval (**GeAR**), which not only improves the global document-query similarity through contrastive learning, but also integrates well-designed fusion and decoding modules. This enables GeAR to generate relevant context within the documents based on a given query, facilitating learning to retrieve local fine-grained information. Furthermore, when used as a retriever, GeAR does not incur any additional computational cost over bi-encoders. GeAR exhibits competitive retrieval performance across diverse scenarios and tasks. Moreover, qualitative analysis and the results generated by GeAR provide novel insights into the interpretation of retrieval results. The code, data, and models will be released to support future research.

1 Introduction

Document retrieval serve as the foundational technology behind large-scale information systems, playing a crucial role in applications such as web search, open-domain question answering (QA) (Chen et al., 2017; Karpukhin et al., 2020), and retrieval-augmented generation (RAG) (Lewis et al., 2020; Liu et al., 2024a; Gao et al., 2024). The predominant approach in passage retrieval is to construct a bi-encoder model (Reimers and Gurevych, 2019). In this framework, queries and documents are encoded separately, converting each into vector representations that enable computation of their semantic similarity in a high-dimensional space.

However, this similarity calculation process faces several challenges. First, the complex semantic relationship between query and document is mapped to a scalar similarity, which cannot reflect enough information and is difficult to understand (Brito and Iser, 2023). Second, when dealing with long documents, such as those with 256, 512, or even more tokens, identifying the section most relevant to the query and contributing most to the similarity is highly desirable but challenging to achieve (Luo et al., 2024; Günther et al., 2024). Moreover, many NLP tasks, such as sentence selection, search result highlighting, needle in a haystack (Liu et al., 2024b; An et al., 2024; Wang et al., 2024), and fine-grained citations (Gao et al., 2023; Zhang et al., 2024), require a deep and fine-grained understanding of the text. Given this need for fine-grained understanding, the bi-encoder that simply aligns the full document to the query seems insufficient, as its conventional contrastive loss mainly emphasizes global semantics (Khattab and Zaharia, 2020). To complement this core capability of the retriever, we propose a novel and challenging fundamental question: *How to make the retriever have both* ***global*** *and* ***local*** *understanding and retrieval capabilities?*

Although the concept is intutive, several challenges remain. First, it is difficult to construct sufficient data to support effective solutions to this problem in previous research work. Second, the training objectives, model architectures, design details, as well as how to effectively train the models, have not been fully explored. To address these challenges, we propose a novel approach **GeAR** (**Ge**neration-**A**ugmented **R**etrieval). In it, we build a pipeline to efficiently synthesize large amounts of high-quality (query-document-information) triples by utilizing large language models. In terms of method, GeAR retains to leverage contrastive learning to optimize the similarity between the query and the global document. To improve the interaction between local information and queries, we design a text decoder that generates fine-grained information from the document in response to a given query. This enhances the model's ability to understand local semantics. In this way, GeAR can handle both the retrieval of global documents and local information simultaneously.

We conduct extensive experiments on two retrieval tasks, and compared with the suboptimal methods, GeAR achieves 3.5% and 12.9% relative improvements on global document retrieval and local information retrieval tasks respectively. GeAR's versatility and visual analysis also shed new light on the interpretability and comprehensibility of retrieval results.

Overall, our contributions are summarized as follows:

- We introduce a new global-local retrieval task, which presents challenges for both document retrieval and fine-grained information retrieval within documents.
- We introduce GeAR, which augmented the model's global and local understanding and retrieval capabilities of documents by incorporating a generation task.
- Through extensive experiments, GeAR has shown competitive performance across various retrieval tasks. GeAR's versatility also makes the retrieval results more explainable.

2 Related Work

2.1 Embedding-based Retrieval

Embedding-based retrieval has emerged as a cornerstone of modern information retrieval systems, enabling efficient semantic search through dense vector representations. Early approaches like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) demonstrated the potential of learning distributed word representations, while more recent transformer-based models such as BERT (Devlin et al., 2019) have pushed the boundaries of contextual embeddings. Bi-encoder architectures (Reimers and Gurevych, 2019) have become particularly popular for retrieval tasks (Huang et al., 2013). Recent advances include contrastive learning objectives (Karpukhin et al., 2020; Wang et al., 2022; Li et al., 2023; Gao et al., 2021) and hard negative mining strategies (Xiong et al., 2021) to improve embedding quality. Muennighoff et al. (2024) explored how to generate text and provide excellent semantic representation by distinguishing task instructions. Multimodal information retrieval also relies on high-quality semantic representations, where the embedding space serves to bridge different modalities, including text, images, and video. Vision language models such as CLIP (Radford et al., 2021), ALBEF (Li et al., 2021), and BLIP (Li et al., 2022) have demonstrated remarkable zero-shot capabilities by learning joint embeddings derived from large scale image-text pairs.

2.2 Fine-grained Information Mining

Mining fine-grained information in a long context during retrieval has become a key challenge for efficient information retrieval. The naive heuristic hierarchical approach involves further chunking documents and then calculate semantic similarity with the query on the chunked sentences. However, finer chunking easily leads to increased computational complexity and semantic incoherence (Yang et al., 2016; Liu et al., 2021; Arivazhagan et al., 2023). In question-answering tasks, RNN or BERT is often used to compute token representations and train classifiers for information extraction (Seo, 2016; Wang, 2016; Chen et al., 2017; Xu et al., 2019). With the development of generative models, there have been many efforts to enhance the model's ability to find a needle in a haystack (Liu et al., 2024b; An et al., 2024; Wang et al., 2024). Another similar task is to have the model add reference information to the original text when generating responses (Gao et al., 2023; Zhang et al., 2024). Coincidentally, some recent research is dedicated to improving the region-level understanding ability of multimodal large language models (MLLMs) (Chen et al., 2024).

Despite these advances, we find that these works often rely on heavy decoder-only models that are independent of the retrieval model, but few focus on mining fine-grained information during the retrieval stage.

3 Generation Augmented Retrieval

3.1 Preliminaries

In this work, we formalize the global-local retrieval task as follows: Let a document corpus as $\mathbb{D}$, which contains N documents $\{d_1, ..., d_i, ..., d_N\}$. Each

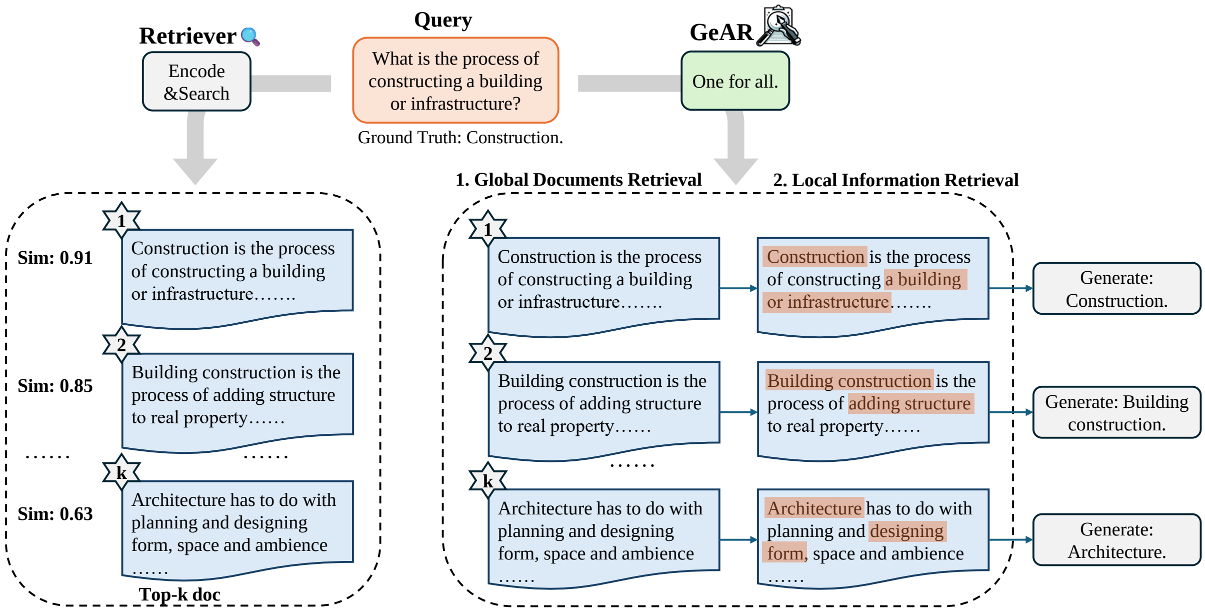


Figure 1: Comparison of functionality between classical retriever and GeAR. GeAR is designed to handle both global document retrieval and local information retrieval simultaneously. In addition, GeAR can generate information based on the query for reference.

of these documents d_i contains a number of fine-grained information units $\{u_1, ..., u_{l_i}\}$, such as sentences, where l_i is the units number of d_i. Our goal is to find a retrieval method $f(\cdot)$, which can retrieve the relevant document d from $\mathbb{D}$, as well as the fine-grained information u from d given query q:

$$f(q, \mathbb{D}) \rightarrow \{d\} \tag{1}$$

$$f(q, d) \rightarrow \{u\} \tag{2}$$

In this work, we explicitly define the process as two tasks, **(1)** the global document retrieval and **(2)** the local information retrieval, as shown in Figure 1.

3.2 Data construction

In this work, we consider two main retrieval scenarios: Question Answer Retrieval (QAR) and Relevant Information Retrieval (RIR). In the following sections, we introduce how the data is constructed and outline the specific goals of the retrieval tasks in each scenario.

Question Answer Retrieval In this scenario, the query q is in the form of a question, and the goal is to retrieve (1) the reference documents d that support answering the question and (2) the fine-grained sentences u that contain the answer.

Relevant Information Retrieval This scenario closely mirrors typical user behavior when searching for information on search engines. The query q is typically a few phrases or keywords, the objective is to retrieve (1) the documents d that correspond to the query and (2) the fine-grained sentences u in the documents that are most relevant to the query. However, a significant challenge in this scenario is the difficulty of collecting suitable data from existing public datasets to address this problem. To overcome this, we construct a pipeline to synthesize high quality data using a large language model. Specifically, we select high quality Wikipedia documents (Foundation), from which we sampled sentences of appropriate length and whose subject is not a pronoun as u. Then we leverage LLM to rewrite these sentences as queries q. After applying de-duplication and relevance filtering, we obtain a promising set of **5.8M** triples. Kindly refer to Appendix A for details on complete data processing procedure.

3.3 Model Structure

This section introduces the architecture of GeAR. It is our intention to enable the model to have both global and local text retrieval capabilities. Inspired by advances in multimodal representation learning (Li et al., 2021, 2022; He et al., 2020), we revisit the task from the perspective of modality alignment. Documents and queries can be regarded as two modalities. We facilitate semantic alignment between documents and queries via a bi-encoder,

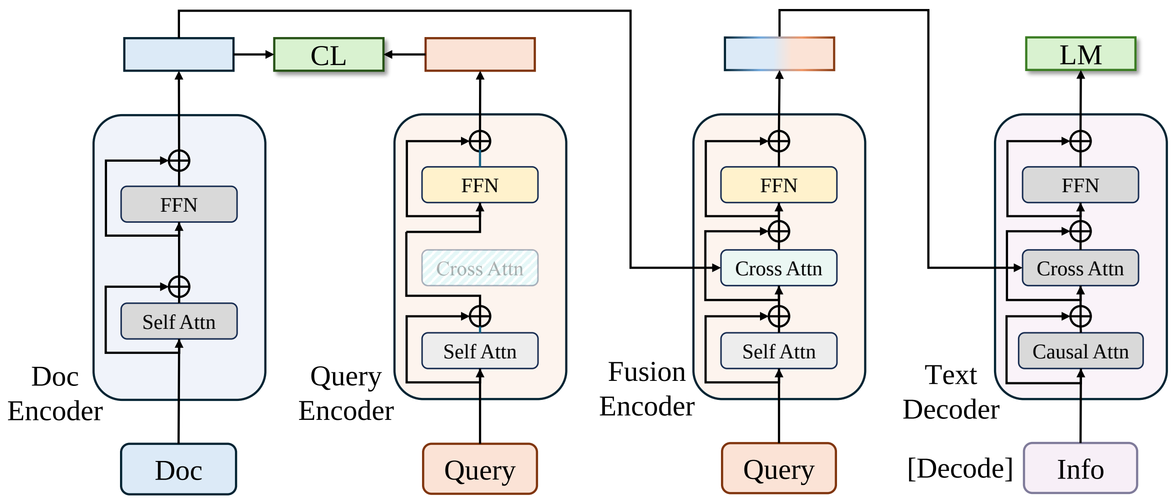


Figure 2: **GeAR.** It consists of a bi-encoder, a fusion encoder, and a text decoder. It contains two training objectives, CL represents contrastive learning loss, which aims to optimize the similarity between documents and queries. LM represents the language modeling loss for generating relevant information given documents and queries.

and enable the model to learn to focus on fine-grained query-related information in documents via a fusion encoder and a generation task. The overview of the GeAR structure is illustrated in Figure 2.

Bi-Encoder In the same setup as the classical retrieval approach, we initialize two encoders $E_d(\cdot)$ for documents and $E_q(\cdot)$ for queries. We use mean pooling to obtain the text embedding.

Fusion Encoder The fusion encoder share most of the parameters with query encoder, but have an lightweight learnable cross attention module. In this part, the document embeddings from $E_d(\cdot)$ are fused with the query embeddings through cross attention at each layer of the fusion encoder.

Text Decoder The text decoder receives the fusion embeddings and generates fine-grained information[1] in the document based on the given query and document. It uses a unidirectional causal attention instead of a bidirectional self-attention. A specific [Decode] token is added to identify the beginning of the sequence. The subsequent auto-regressive decoding process will interact with the generated tokens and fusion embeddings to generate text.

3.4 Training Objectives

In this section, we introduce the training objectives of GeAR. Through the joint modeling of natural language understanding and natural language generation, GeAR can handle global document retrieval and local information retrieval simultaneously.

Contrastive Learning Loss (CL) We use bi-encoder to encode the queries and documents, and optimize the semantic similarity between them through contrastive learning loss (CL). In addition, we followed the practice in MoCo (He et al., 2020) and BLIP (Li et al., 2022), where a momentum Bi-Encoder is introduced to encode momentum embeddings and provide richer supervised signals as soft labels.

Language Modeling Loss (LM) The introduction of LM loss is crucial for enhancing the local information retrieval capability of GeAR. LM activates the text decoder, enabling the model to generate relevant intrinsic information by leveraging the fusion embeddings of document and query. It guides the model to learn the fine-grained semantic fusion between query and document. LM optimizes the cross-entropy loss over the entire vocabulary, maximizing the likelihood of the ground truth text. The overall loss of GeAR is the sum of $\mathcal{L}_{\mathrm{CL}}$ and $\mathcal{L}_{\mathrm{LM}}$ with a optional weight α:

$$\mathcal{L}_{\mathrm{GeAR}} = \mathcal{L}_{\mathrm{CL}} + \alpha * \mathcal{L}_{\mathrm{LM}} \tag{3}$$

3.5 Inference

GeAR's inference process is flexible. In this section, we introduce various usages of GeAR to accomplish different tasks.

Global Documents Retrieval For this task, we can use the bi-encoder part of GeAR to compute

[1]Note that in the QAR scenario, the ground truth for the generation is the answer itself, not the full sentence u in which answer appears.

the similarity between query and document like the previous classic retrieval method, without introducing any additional parameters and computation cost.

Local Information Retrieval The fusion encoder in GeAR interacts query and document via cross attention. The cross attention weights between each sentence in the document and the query reflect which information the model prioritizes. We rank the sentences based on this weights to retrieve the most relevant fine-grained information from the document.

4 Experiments

In this section, we first outline the experimental setup, and then we discuss the overall performance of each task and more detailed analysis.

4.1 Setup

Datasets For Question Answer Retrieval, we sampled 30M data from PAQ (Lewis et al., 2021) datasets to train GeAR, and sampled 1M documents and 20k queries as test set. To verify the generalization ability of methods, we also evaluate the performance on three additional held-out datasets: SQuAD (Rajpurkar et al., 2016), NQ (Kwiatkowski et al., 2019), and TriviaQA (Joshi et al., 2017). For Relevant Information Retrieval, we leverage the synthesized 5.8M data, of which 95% is used for training and 5% is reserved for the test set. Specific dataset statistics are in Appendix B.

Training Details "bert-base-uncased" (Devlin et al., 2019) is used to initialize the encoders in GeAR. The decoder also has 110M parameters, but is randomly initialized. We train GeAR for 10 epochs using batch size of 48 (QAR) / 16 (RIR) on 16 AMD MI200 GPUs. We set the weight $\alpha = 0.25$. We use the AdamW (Loshchilov, 2017) optimizer with a weight decay of 0.05. The full hyperparameters and training settings are detailed in Appendix C.

Baselines We compare GeAR with two types of baselines, one is the text embedding models that have been adequately pre-trained on a large corpus, including SBERT (Reimers and Gurevych, 2019), E5 (Wang et al., 2022), BGE (Xiao et al., 2024), and GTE (Li et al., 2023). The models involved in the comparison are all base versions. Since the training data of the pre-trained model partially overlaps with the evaluation data, their performance are used as an important reference. To ensure a fairer comparison, we retrain SBERT[2] (Reimers and Gurevych, 2019) and BGE[3] (Xiao et al., 2024) using the open sourced training pipelines with aligned training data and initialization, referred to as SBERT_{RT} and BGE_{RT} in the following. In addition, we also compare with the more complex BGE reranker Base and Large (Xiao et al., 2024) in the Local Information Retrieval task. In the section 4.2, we underline the best performance of the pre-trained models and bold the best performance of the retrained models.

4.2 Overall performance

In this section, we present the overall performance on global Documents Retrieval and local information retrieval.

Global Documents Retrieval Firstly, Table 1 reports the comparison with existing methods on global documents retrieval task. We find that GeAR delivers competitive performance across multiple datasets even with only tens of millions of training data, demonstrating efficient data utilization. As a reference, the pre-trained SBERT model used 1.17B sentence pairs. GeAR achieves the state-of-the-art performance on the three datasets SQuAD, PAQ, and RIR, and is slightly weaker than the pre-trained GTE on the NQ dataset. It only lags behind significantly on TriviaQA, but is also better than E5. At the same time, GeAR outperforms the retrained model in all metrics. Compared with BGE_{RT}, GeAR achieves relative improvement of 3.5% in average Recall@5, highlighting the effectiveness of our training method. In Section 4.3, we further discuss the role of the generation task and its effect on model performance.

Local Information Retrieval Next, we evaluate the performance of each method on the local information retrieval task. In the evaluation process, we provide the query and the document (q, d) to the model and observe whether it is able to retrieve the corresponding fine-grained unit u. For the retrieval model, we split the documents into sentences and compute their similarity to the query independently, selecting the top-k sentences. In contrast, GeAR retrieves units based on the cross attention weights for each sentence given the query, as described in Section 3.5. The results are reported in Table 2.

It is observed that SBERT_{RT} and BGE_{RT} perform mediocrely, as their training objective focus

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[3] https://github.com/FlagOpen/FlagEmbedding

Method	SQuAD		NQ		TriviaQA		PAQ		RIR	
	R@5	M@5	R@5	M@5	R@5	M@5	R@5	M@5	R@5	M@5
Pre-trained retrieval model										
SBERT	0.812	0.667	0.754	0.576	0.677	0.413	0.808	0.701	0.376	0.297
E5	0.803	0.674	0.760	0.581	0.645	0.390	0.816	0.716	0.484	0.396
BGE	0.829	0.701	0.674	0.502	0.690	0.422	0.752	0.647	0.451	0.367
GTE	0.866	0.744	0.767	0.587	0.726	0.443	0.836	0.736	0.528	0.435
Retrained retrieval model										
SBERT_{RT}	0.742	0.585	0.739	0.550	0.577	0.342	0.859	0.742	0.739	0.631
BGE_{RT}	0.841	0.701	0.751	0.553	0.640	0.384	0.901	0.802	0.953	0.881
GeAR	0.887	0.766	**0.762**	**0.574**	**0.664**	**0.400**	0.952	0.872	**0.964**	**0.910**
$\text{GeAR}_{w/o\mathcal{L}_{\text{LM}}}$	**0.889**	**0.776**	0.755	0.565	0.660	0.399	**0.955**	**0.877**	0.963	0.907

Table 1: Comparison of global documents retrieval performance on different datasets, where R@k stands for Recall@k, M@k stands for MAP@k.

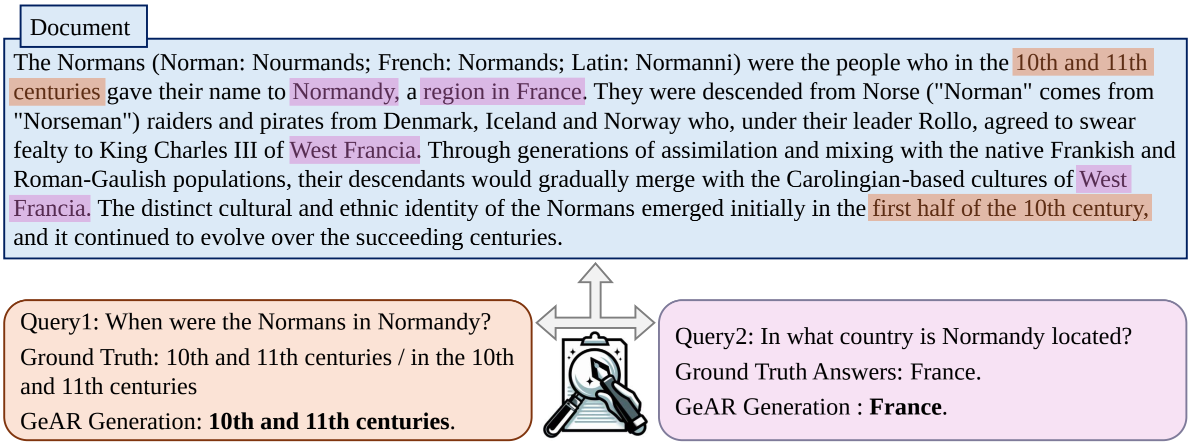
Method	SQuAD		NQ		TriviaQA		PAQ		RIR	
	R@1	M@1	R@1	M@1	R@1	M@1	R@1	M@1	R@3	M@3
Pre-trained retrieval model										
SBERT	0.739	0.800	0.558	0.652	0.359	0.583	0.498	0.561	0.891	0.874
E5	0.783	0.847	0.590	0.683	0.379	0.613	0.573	0.640	0.891	0.878
BGE	0.768	0.830	0.570	0.663	0.362	0.589	0.565	0.630	0.894	0.881
GTE	0.758	0.820	0.548	0.639	0.352	0.572	0.525	0.590	0.895	0.886
Retrained retrieval model										
SBERT_{RT}	0.516	0.568	0.445	0.523	0.281	0.472	0.363	0.418	0.899	0.881
BGE_{RT}	0.455	0.538	0.601	0.656	0.288	0.475	0.409	0.466	0.897	0.888
Reranker model										
BGE-Reranker-B	0.690	0.749	0.641	0.740	0.399	0.640	0.690	0.762	0.884	0.850
BGE-Reranker-L	0.751	0.813	0.670	0.770	0.464	0.737	0.704	0.778	0.891	0.873
GeAR	**0.814**	**0.878**	**0.761**	**0.865**	**0.510**	**0.797**	**0.884**	**0.965**	**0.933**	**0.897**
$\text{GeAR}_{w/o\mathcal{L}_{\text{LM}}}$	0.803	0.869	0.582	0.677	0.402	0.650	0.649	0.720	0.891	0.886

Table 2: Comparison of local information retrieval performance on different datasets, where R@k stands for Recall@k, M@k stands for MAP@k.

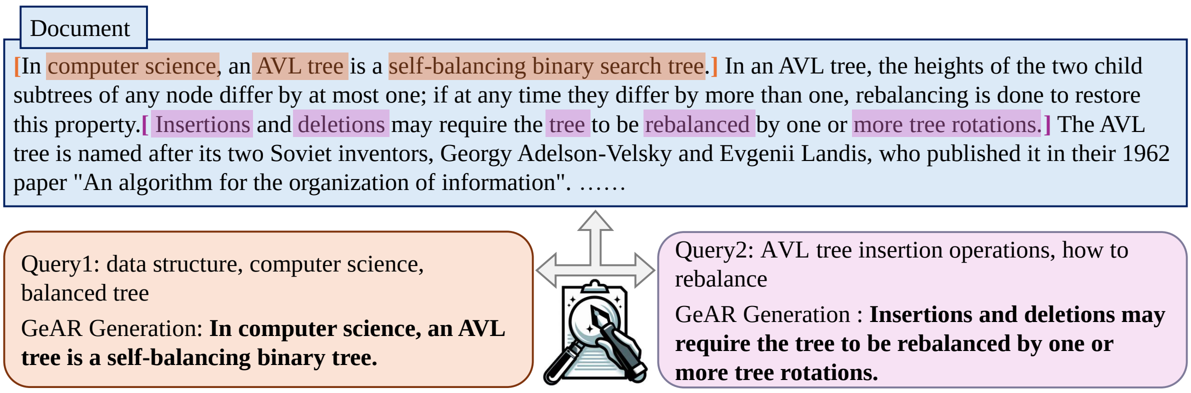
solely on optimizing the overall similarity between the document and the query, neglecting the fine-grained semantic relationships. The more complex BGE-reranker model performs better than the pure retrieval model. GeAR leads the way in all metrics, showing an average relative improvement of 12.9% over the suboptimal BGE-Reranker-L. Notably, GeAR does not require further chunking and encoding of the document. In contrast, GeAR benefits from the joint end-to-end training of retrieval and generation, enabling it not only retrieve documents closely aligned with the query but also effectively retrieve fine-grained information within the document.

4.3 Analysis

The Effect of Language Modeling Objectives In this work, we not only optimize the retrieval performance through contrastive learning, but also enhance GeAR through the information generation task of a given query, so that it has fine-grained semantic understanding and content retrieval ca-



(a) Local information retrieval and generation results of GeAR in Question Answer Retrieval scenario.



(b) Local information retrieval and generation results of GeAR in Related Information Retrieval scenario. The sentences in brackets of corresponding colors are the ground truth of the query.

Figure 3: Visualization of local information retrieval of GeAR . In the two scenarios, we pose two different queries for each document and highlight the top 10 tokens with the highest cross attention weights. The tokens with orange background are for query1 , with purple background are for query2 . We also show the generated results of GeAR.

pabilities. We find that if LM loss is removed, both global and local retrieval performance of the model is reduced, as shown in the last row of Table 1 and Table 2. Further, we also explore the impact of the weight of LM loss on the overall performance. In Table 4, we observed that the effect of the generation on the retrieval performance is inverted U-shaped, with the optimal values at 0.25 and 0.5 respectively. Higher weights may cause the model to focus on learning the generation task instead, which is similar to previous findings (Sener and Koltun, 2018).

Visualization of Local Information Retrieval The key distinction between GeAR and traditional retriever is its ability to mine the local information within the document that is most relevant to the query. Figure 3 illustrates this process and the generation results of GeAR across different scenarios. For each document, we provide two distinct queries and highlight the top 10 tokens with the highest cross attnetion weights corresponding to each query. In Figure 3(a), the two queries are related to time and location respectively. GeAR not only provides the correct answers but also dynamically adjusts its query-specific focus: it assigns higher attention weights to time-related tokens to the first query and prioritizes tokens related to countries and regions to the second query. In Figure 3(b), depending on the query, GeAR focuses on the concept of AVL tree, as well as operations such as insertion and rebalancing, generating corresponding sentences. It is evident that the added generation task enhances the accuracy of local information retrieval. Furthermore, GeAR can not only retrieve detailed content related to the query but also generates corresponding text for reference. This advancement shifts the retrieval results from being mere numerical values to more intuitive and explainable.

Method	SQuAD		NQ		TriviaQA		PAQ		RIR	
	EM	F1	EM	F1	EM	F1	EM	F1	Rouge-1	Rouge-L
Llama 3.2 3B	60.7	73.3	57.7	59.9	50.4	66.7	62.7	75.5	69.4	67.9
Llama 3.3 70B	**66.2**	**77.7**	61.0	**66.9**	**56.6**	**73.0**	61.0	74.5	84.4	84.0
GeAR	60.0	64.5	**65.7**	60.7	46.5	59.1	**87.5**	**91.9**	**87.6**	**87.3**

Table 3: Generation performance on different datasets.

α	Global Retrieval	Local Retrieval
	Ave Recall	Ave Recall
0	0.844	0.663
0.25	**0.846**	0.781
0.5	0.844	**0.785**
0.75	0.839	0.784
1	0.838	0.784

Table 4: Comparison of performance on two retrieval tasks when the LM loss weight α is varied.

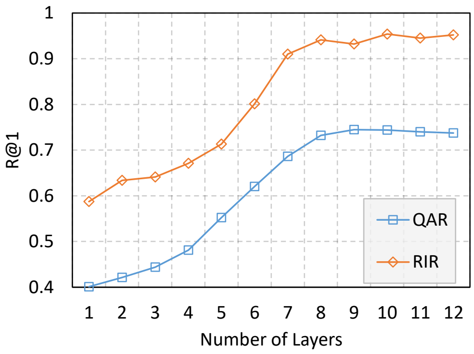


Figure 4: Local information retrieval performance of different layers.

Local Retrieval Performance of Different Layers In GeAR, the query and document tokens interact through the cross attention module at each layer of the fusion encoder. In Figure 4, we plot the local retrieval performance using cross attention weights across different layers to examine its relationship with model depth. The results indicate that higher layers generally perform well, as the token embeddings at these layers capture rich semantic information. Interestingly, we observe that the highest layer does not yield the best performance. Instead, peak performance is reached in the last 3 to 4 layers[4]. This phenomenon may arise due to the representations in the highest layer are optimized to serve the final task rather than intermediate interactions. Similar observations have been reported in previous studies involving encoder-only and decoder-only models (Jawahar et al., 2019; Skean et al., 2024).

[4]In this work, we utilized the 10th layer.

Information Generation Although generation serves as an auxiliary task in GeAR and the decoder is lightweight, we are nonetheless interested in its generation performance. Table 3 reports the Exact Match (EM) and F1 scores on the QA datasets, and the Rouge (Lin, 2004) scores on the RIR dataset. For reference, we include results from the Llama series model (Dubey et al., 2024). Notably, GeAR achieves surprising performance on the in-domain data, and performs reasonably well on other test sets. Additionally, Figure 3 illustrates examples of GeAR's ability to generate answers and relevant information, showcasing its satisfactory generation capabilities.

5 Conclusion

In this work, to address the challenges of unexplainable and coarse-grained results inherent in current bi-encoder retrieval methods, we propose a direct and effective modeling method: **Ge**neration **A**ugmented **R**etrieval (GeAR). GeAR enhances fine-grained information retrieval by introducing a generation task and incorporating a lightweight decoder and cross attention module, while maintaining the efficiency of the bi-encoder. Experimental results across multiple retrieval tasks and two different scenarios demonstrate that GeAR achieves excellent performance and have both global and local understanding and retrieval capabilities. Qualitative analysis further highlights its intuitive and explainable retrieval results. These capabilities make GeAR particularly promising in downstream tasks such as web search and retrieval-augmented generation (RAG). We hope that this work offers valuable insights into the gradual unification of natural language understanding and generation paradigms, paving the way for more general and explainable retrieval systems in the future.

Limitations

Due to constraints in computational resources and associated costs, the synthesized data used in our experiments is not as comprehensive as that found in traditional retrieval scenarios. While the results demonstrate the efficacy of GeAR, applying it to more diverse and semantically rich retrieval scenarios remains an important direction for future exploration.

Additionally, the context length of GeAR is limited to 512 tokens, consistent with the chunk lengths commonly used in retrieval tasks. However, recent advancements in extending the context length of retrieval models, such as those proposed in (Zhu et al., 2024), suggest exciting opportunities to overcome this limitation. Extending GeAR's context length could further enhance its capabilities in handling long-form retrieval tasks, which we plan to investigate in future work.

Thirdly, the decoder of GeAR has only 110M parameters, the same as the encoder. Moreover, the focus of GeAR is not to optimize the generation performance of the model, and the generation task is not the main task. Therefore, GeAR cannot complete other complex generation tasks like Llama (Dubey et al., 2024). In future work, whether GeAR can be scaled up to enable it to complete retrieval tasks and respond well to various generation problems will be an interesting direction.

We hope that the above discussions can inspire further investigation within the research community, encouraging advancements that address these limitations and contribute to the broader progress of NLP research.

References

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. 2024. Make your llm fully utilize the context. In *NeurIPS 2024*.

Manoj Ghuhan Arivazhagan, Lan Liu, Peng Qi, Xinchi Chen, William Yang Wang, and Zhiheng Huang. 2023. Hybrid hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10680–10689, Toronto, Canada. Association for Computational Linguistics.

Eduardo Brito and Henri Iser. 2023. Maxsime: Explaining transformer-based semantic similarity via contextualized best matching token pairs. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2154–2158, New York, NY, USA. Association for Computing Machinery.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Hong-You Chen, Zhengfeng Lai, Haotian Zhang, Xinze Wang, Marcin Eichner, Keen You, Meng Cao, Bowen Zhang, Yinfei Yang, and Zhe Gan. 2024. Contrastive localized language-image pre-training. *arXiv preprint arXiv:2410.02746*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Wikimedia Foundation. Wikimedia downloads.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

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.

Michael Günther, Isabelle Mohr, Daniel James Williams, Bo Wang, and Han Xiao. 2024. Late chunking: contextual chunk embeddings using long-context embedding models. *arXiv preprint arXiv:2409.04701*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

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.

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

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.

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

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.

Patrick Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pretraining for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

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

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024a. se^2: Sequential example selection for in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5262–5284, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip Yu. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200, Punta Cana, Dominican Republic. Association for Computational Linguistics.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Kun Luo, Zheng Liu, Shitao Xiao, and Kang Liu. 2024. Bge landmark embedding: A chunking-free embedding method for retrieval augmented long-context large language models. *arXiv preprint arXiv:2402.11573*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

M Seo. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. 2024. Does representation matter? exploring intermediate layers in large language models. *arXiv preprint arXiv:2412.09563*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, Yunshui Li, Min Yang, Fei Huang, and Yongbin Li. 2024. Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646, Miami, Florida, USA. Association for Computational Linguistics.

S Wang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations (ICLR)*.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Jiajie Zhang, Yushi Bai, Xin Lv, Wanjun Gu, Danqing Liu, Minhao Zou, Shulin Cao, Lei Hou, Yuxiao Dong, Ling Feng, et al. 2024. Longcite: Enabling llms to generate fine-grained citations in long-context qa. *arXiv preprint arXiv:2409.02897*.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. LongEmbed: Extending embedding models for long context retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 802–816, Miami, Florida, USA. Association for Computational Linguistics.

Appendices

A Data Construction

We present here the practice of synthesizing data for Relevant Information Retrieval scenarios.

Pre-processing Firstly, we choose high-quality documents from Wikipedia (Foundation). We process the documents sentence by sentence, removing sentences with repetitive line breaks and phrases, until the document processing is complete or the token count reaches 500 (<512). We remove the documents that are too short, with a sentence count less than 3 or a token count of less than 200. Second, we filter the candidate sentences in the document that can be rewritten: we filter all the sentences that have a token count between 8 and 20 and whose first word and subject are not pronouns (the set of pronouns includes "this", "these", "it", "that", "those", "they", "he", "she", "we", "you", "I"). If the number of sentences filtered is less than 3, we discard the document.

LLM Rewriting We randomly select 3 sentences in the document and use vLLM (Kwon et al., 2023) and "Llama-3.1-70B-Instruct" (Dubey et al., 2024) to rewrite them into queries, the prompt is: "You are a helpful assistant, please help the user to complete the following tasks directly, and answer briefly and fluently. This is a sentence from Wikipedia. Assuming that users want to search for this sentence on a search engine, write a phrase that users might use to search (including some keywords), separated by commas. Retain the key information of the subject, object, and noun. Unimportant words can be modified, but do not add other information.".

Post-processing We de-duplicate the keywords in the rewritten query and then reorder them. To ensure the relevance of the query to the document, we perform a round of filtering using BGE (Xiao et al., 2024) to retain the data with a similarity of 0.5 or more between the rewritten query and the document. In this way we obtain a reasonable triad of queries, documents, and units (sentences).

For the construction of Relevant Information Retrieval data, we have also tried to collect paired sentences and make LLM expand one of them into a document. However, we fine that other sentences in the LLM expansion were less informative than the original sentence, for example, being some descriptive statements were generated around the original sentence. This pattern tends to cause the model to learn to locate the central sentence, or the most informative sentence, in the expanded document, leading model to ignore the query. So please be aware of this if you plan to try this way of constructing your data.

Hyperparameter	**Assignment**
Computing Infrastructure	16 MI200-64GB GPUs
Number of epochs	10
Batch size per GPU	48 / 16
Maximum sequence length	512
Optimizer	AdamW
AdamW epsilon	1e-8
AdamW beta weights	0.9, 0.999
Learning rate scheduler	Cosine lr schedule
Initialization learning rate	1e-5
Minimum learning rate	1e-6
Weight decay	0.05
Warmup steps	1000
Warmup learning rate	1e-6

Table 5: Hyperparameter settings

B Overview of datasets

We describe here in detail the datasets used for training and evaluation.

B.1 Training

For Question Answer Retrieval, we sampled 30M data from PAQ (Lewis et al., 2021) datasets to train GeAR. For Relevant Information Retrieval, we used the 95% of the synthetic data for training. The specific statistics are shown in Table 6.

Scenario	Data Number
QAR	30,000,000
RIR	5,676,877

Table 6: Training data statistics.

B.2 Evaluation

In the evaluation stage, we introduce the specific information of the evaluation data by task.

Global Documents Retrieval First, for the global document retrieval task, the queries come from the test set in the respective dataset, and the candidate documents are all documents within the entirety of the dataset, including the SQuAD (Rajpurkar et al., 2016), NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and RIR

Scenario	Dataset	Documents Number	Queries Number
QA	Squad	20,239	5,928
	NQ	64,501	2,889
	TriviaQA	104,160	14,000
	PAQ	932,601	20,000
RIR	RIR	2,315,413	145,562

Table 7: The evaluation data statistics for the global document retrieval task.

Scenario	Dataset	Data Number
QA	Squad	5,928
	NQ	2,889
	TriviaQA	14,000
	PAQ	20,000
RIR	RIR	10,000

Table 8: The evaluation data statistics for the local information retrieval and generation tasks.

datasets. It is difficult to encode all the documents of the PAQ dataset because the dataset is too large. So for the PAQ dataset, we sampled 1M documents and 20k queries, all of which have no intersection with the training data. The evaluation data statistics for the document retrieval task are shown in Table 7.

Local Information Retrieval and Generation For these two tasks, we directly use the test set data corresponding to the respective datasets. Therefore, their number is consistent with the number of queries in Table 7. For the RIR dataset, we sample 10k records as the test set. The evaluation data statistics for the local information retrieval and generation tasks are shown in Table 8.

C HyperParameters and Implementation Details

We run model training on 16 AMD MI200 GPUs with 64GB memory and evaluation on 8 NVIDIA Tesla V100 GPUs with 32GB memory. The learning rate is warmed-up from $1e$-6 to $1e$-5 in the first 1000 steps, and then following a cosine scheduler, where the mininum learning rate is $1e$-6. The momentum parameter for updating momentum encoder is set as 0.995, the queue size is set as 57600. We linearly ramp-up the soft labels weight from 0 to 0.4 within the first 2 epoch. The overall hyperparameters are detailed in Table 5. We use FAISS (Douze et al., 2024; Johnson et al., 2019) to store and search for vectors. The 2 encoders and 1 decoder in GeAR are the same size as "bert-base" (Devlin et al., 2019), the total number of parameters of GeAR is about 330M. The training time for QAR scenario is about 5 days, for RIR scenario is about 3 days.

D More Visualization

To present the effect of GeAR intuitively, we show more visualisation results of GeAR in Figure 5. Each example contains two different queries for a document to observe whether GeAR can respond differently to different queries, including locating key information and generating answers. We also highlight the top 10 tokens with the highest cross attention weights for the corresponding queries. The tokens with orange background are for query1, and the tokens with purple background are for query2.

Document

Computational complexity theory is a branch of the theory of computation in theoretical computer science that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

Query1: What branch of theoretical computer science deals with broadly classifying computational problems by difficulty and class of relationship?

Ground Truth: Computational complexity theory

GeAR Generation: **Computational complexity theory.**

Query2: By what main attribute are computational problems classified utilizing computational complexity theory?

Ground Truth Answers: [inherent difficulty, their inherent difficulty]

GeAR Generation : **Inherent difficulty.**

Document

Southern California, often abbreviated SoCal, is a geographic and cultural region that generally comprises California\'s southernmost 10 counties. The region is traditionally described as "eight counties", based on demographics and economic ties: Imperial, Los Angeles, Orange, Riverside, San Bernardino, San Diego, Santa Barbara, and Ventura. The more extensive 10-county definition, including Kern and San Luis Obispo counties, is also used based on historical political divisions. Southern California is a major economic center for the state of California and the United States.

Query1: What is a major importance of Southern California in relation to California and the United States?

Ground Truth: [economic center, major economic center]

GeAR Generation: **Major economic center.**

Query2: What are the ties that best described what the "eight counties" are based on?

Ground Truth: [demographics and economic ties, economic, demographics and economic]

GeAR Generation : **Demographics and economic ties.**

Document

Formed in November 1990 by the equal merger of Sky Television and British Satellite Broadcasting, BSkyB became the UK's largest digital subscription television company. Following BSkyB's 2014 acquisition of Sky Italia and a majority 90.04% interest in Sky Deutschland in November 2014, its holding company British Sky Broadcasting Group plc changed its name to Sky plc. The United Kingdom operations also changed the company name from British Sky Broadcasting Limited to Sky UK Limited, still trading as Sky.

Query1: What is the name of the holding company for BSkyB?

Ground Truth: [Sky plc, British Sky Broadcasting Group plc, British Sky Broadcasting Group plc]

GeAR Generation: **British sky broadcasting group plc.**

Query2: What year did BSkyB acquire Sky Italia?

Ground Truth: 2014.

GeAR Generation : **2014.**

Document

In November 2006, the Victorian Legislative Council elections were held under a new multi-member proportional representation system. The State of Victoria was divided into eight electorates with each electorate represented by five representatives elected by Single Transferable Vote. The total number of upper house members was reduced from 44 to 40 and their term of office is now the same as the lower house members—four years. Elections for the Victorian Parliament are now fixed and occur in November every four years. Prior to the 2006 election, the Legislative Council consisted of 44 members elected to eight-year terms from 22 two-member electorates.

Query1: What kind of representational system does the Victorian Legislative Council have?

Ground Truth: [multi-member proportional, multi-member proportional representation system]

GeAR Generation: **Multi-member proportional representation system.**

Query2: How often are elections held for the Victorian Parliament?

Ground Truth: [every four years, four years]

GeAR Generation : **Every four years.**

Figure 5: More Visulization results.