Research Article

GeAR: Generation Augmented Retrieval

Haoyu Liu[1], Shaohan Huang[1], Jianfeng Liu[1], Yuefeng Zhan[1], Hao Sun[1], Weiwei Deng[1], Feng Sun[1], Furu Wei[1], Qi Zhang[1]

1. Microsoft (United States), Redmond, United States

Document retrieval techniques form the foundation for the development of large-scale information systems. The prevailing methodology is to construct a bi-encoder and compute the semantic similarity. However, such scalar similarity is difficult to reflect enough information and impedes our comprehension of the retrieval results. In addition, this computational process mainly emphasizes the global semantics and ignores the fine-grained semantic relationship between the query and the complex text in the document. In this paper, we propose a new method called Generation Augmented Retrieval (GeAR) that incorporates well-designed fusion and decoding modules. This enables GeAR to generate the relevant text from documents based on the fused representation of the query and the document, thus learning to "focus on" the fine-grained information. Also when used as a retriever, GeAR does not add any computational burden over bi-encoders. To support the training of the new framework, we have introduced a pipeline to efficiently synthesize high-quality data by utilizing large language models. GeAR exhibits competitive retrieval and localization performance across diverse scenarios and datasets. Moreover, the 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 after completing technical review to facilitate 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) [1][2], and retrieval-augmented generation (RAG) [3][4][5]. The predominant approach in passage retrieval is to construct a bi-encoder model. In this architecture, queries and documents are encoded separately, transforming each into vector representations that enable computation of their semantic similarity in a high-dimensional space.

doi.org/10.32388/GWQ6LR

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 [6]. 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 [7][8]. Moreover, many NLP tasks, such as sentence selection, search result highlighting, needle in a haystack [9][10][11], and fine-grained citations [12][13], require a deep and fine-grained understanding of the text.

Given this need for fine-grained understanding, the bi-encoder that simply aligns the entire document to the query seems insufficient, as its conventional contrastive loss mainly emphasizes global semantics [14]. To complement this core localization capability of the retriever, we propose a novel and challenging fundamental question: Can we enhance and integrate the information localization capability of existing retrievers without sacrificing their inherent retrieval capabilities?

To address these challenges, we proposed a novel approach **GeAR** (**G**eneration-**A**ugmented **R**etrieval). Specifically, we construct the data into (query-document-information) triples, still using contrastive learning to optimize the similarity between the query and the document. At the same time, we design a text decoder to generate the relevant fine-grained information in the document given the query and document to enhance the retrieval and localization capabilities. Although the concept is simple, there are many challenges. First, it is difficult to find sufficient data to support our solution to this problem in previous research work. Second, the training objectives of retrieval and generation tasks, model architectures, and more design details, as well as how to effectively train the models, have not been fully explored. To this end, we explored a complete pipeline from data synthesis, structure design, to model training. Overall, our contributions are summarized as follows:

- We proposed GeAR, which enhances the model's ability to understand and locate text in a fine-grained manner by jointly modeling natural language understanding and natural language generation. At the same time, the inference process is very flexible to handle different tasks.
- We abstract a new retrieval task that takes into account the problems present in the current retrieval scenario. To solve this task and to support model training, we built a pipeline to synthesize a large amount of high quality data using LLM.
- Through extensive experiments, GeAR has shown competitive performance in retrieval tasks and fine-grained information localization tasks. At the same time, GeAR can also generate relevant

information based on the query and document to help us understand the retrieval results, bringing a new perspective to the traditional retrieval process.

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[15] and GloVe[16] demonstrated the potential of learning distributed word representations, while more recent transformer-based models such as BERT[17] have pushed the boundaries of contextual embeddings. Bi-encoder architectures[18] have become particularly popular for retrieval tasks[19]. Recent advances include contrastive learning objectives[2][20][21][22], hard negative mining strategies[23], and knowledge distillation techniques[24] to improve embedding quality while maintaining computational efficiency.[25] 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[26], ALBEF[27], and BLIP[28] have demonstrated remarkable zero-shot capabilities by learning joint embeddings derived from large scale image-text pairs. These advances made cross-modal retrieval tasks such as text-to-image search and image-to-text retrieval possible.

2.2. Information Localization

Information localization in massive corpora and contents has become a key research direction for improving response accuracy and factual basis. The classic methods used RNN or BERT to compute token representations and trained a classifier for information extraction[29][30][1][31]. The heuristic hierarchical approach involves further chunking the document and then calculating the semantic similarity with the query on the chunked sentences or units for localization. However, finer chunking also results in increased computational complexity and semantic incoherence[32][33][34]. With the development of generative models, there have been many recent efforts to enhance the model's ability to find a needle in a haystack[9][10][11], that is, to locate key information such as sentences in long texts. Another type of similar task is to have the model add reference information to the original text

doi.org/10.32388/GWQ6LR

when generating responses[12][13]. Coincidentally, there have been some recent works focusing on improving the regional level understanding ability of multimodal large language models (MLLMs)[35]. Despite these advances, we have found that there is currently little focus on fine-grained information localization during the retrieval stage.

3. Generation Augmented Retrieval

3.1. Preliminaries

In this work, we formalize the retrieval task with localization as follows: Let a document corpus as $\mathbb{D}$, which contains N documents $\{d_1, \ldots, d_i, \ldots, d_N\}$. Each of these documents d_i contains a number of fine-grained information units $\{u_1, \ldots, u_{l_i}\}$, 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 unit 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 document retrieval task and (2) the fine-grained unit localization task, as Figure 1 showing. It can be seen that the triples of query, document, and unit, represented by the symbols (q, d, u), are fundamental to the definition and resolution of this task.

doi.org/10.32388/GWQ6LR

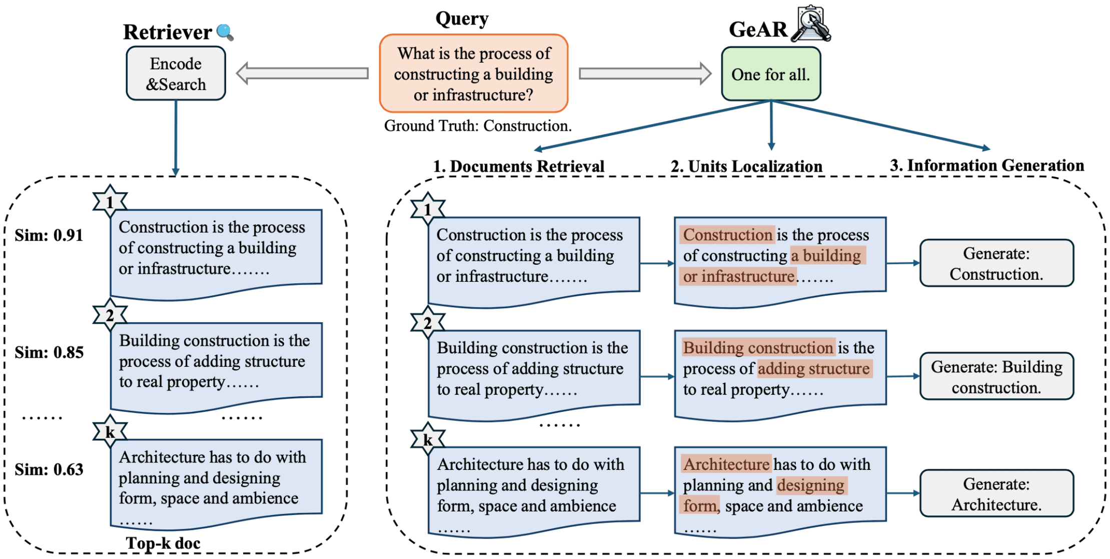

Retriever
Encode &Search
Query
What is the process of constructing a building or infrastructure?
Ground Truth: Construction.
GeAR
One for all.
1. Documents Retrieval
2. Units Localization
3. Information Generation
Sim: 0.91
Sim: 0.85
Sim: 0.63
1
2
k
Construction is the process of constructing a building or infrastructure…….
Building construction is the process of adding structure to real property……
Architecture has to do with planning and designing form, space and ambience ……
Top-k doc
Generate: Construction.
Generate: Building construction.
Generate: Architecture.

Figure 1. Comparison of functionality between classical retriever and GeAR. GeAR is designed to handle both document retrieval and fine-grained unit localization simultaneously, while also generating auxiliary information for reference.

3.2. Data construction

In this work, we focus on two common retrieval scenarios: (1) Question Answer Retrieval (QAR) and (2) Relevant Information Retrieval (RIR). In the following sections, we will introduce how the data are constructed and how they correspond to the triples (q, d, u) mentioned above.

Question Answer Retrieval

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

Relevant Information Retrieval

In this scenario, the query q is in the form of a few phrases or keywords, the objective is to retrieve the documents d that correspond to the query and the fine-grained sentences u in the documents that are most relevant to the query. The scenario is very close to what users normally do when they search for information on the web. The challenge is that we have difficulty in finding suitable data in the current public dataset to drive our problem solving. Therefore, we constructed a pipeline to synthesize high quality data using a large language model. Specifically, we selected high quality Wikipedia documents[36], from which we will sample sentences of appropriate length and whose subject is not a pronoun as u. Then we will leverage LLM to rewrite them as queries q. After de-duplication and relevance filtering, we get promising **5.8M** triples. Kindly refer to Appendix A for details on complete data processing procedure.

3.3. Model Structure

This section presents the architecture of GeAR. It is our intention that the model not only has powerful retrieval capability, but also has the ability to locate key information in documents. Inspired by advances in multimodal representation learning[27][28][37], we revisit the task from a modal alignment perspective. Documents and queries can be considered as two modalities. We facilitate semantic alignment between documents and queries via a bi-encoder, and enable the model to learn to attend to fine-grained query-related information in the document via a fusion encoder and a generation task. The overview of the GeAR structure is illustrated in Figure 2.

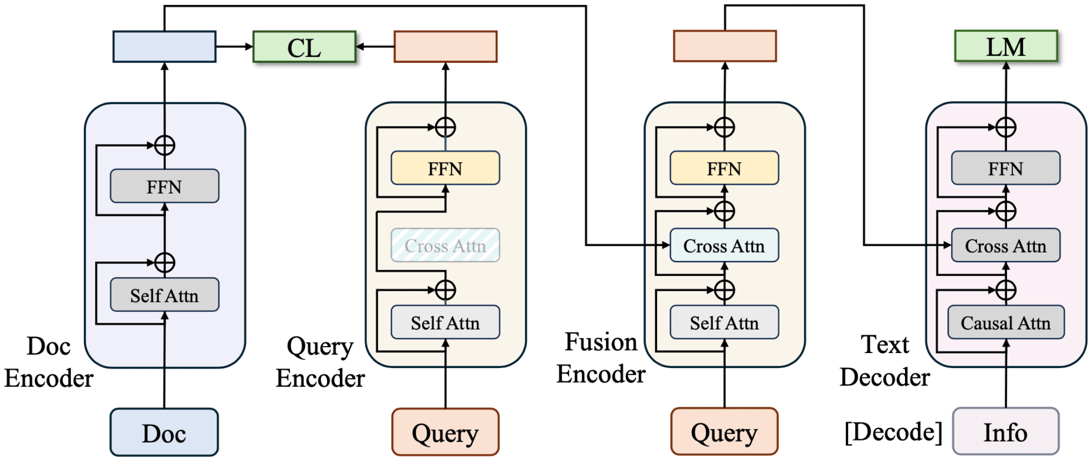


Figure 2. GeAR. It consists of a bi-encoder, a fusion encoder, and a text decoder. It contains two training objectives, CL represents contrastive learning loss, which aims to optimize the similarity between documents and queries. LM represents the language modeling loss for generating relevant information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

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 extra 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 present the training objectives of GeAR. We make the model capable of both retrieval as well as fine-grained semantic understanding and localization through joint natural language understanding and natural language generation modeling.

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[37] and BLIP[28], 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 key to enhancing the information localization capability of GeAR. LM activates the text decoder, which enables the model to generate relevant information using the fusion embeddings of document and query. It guides the model to learn the fine-grained semantic fusion of 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 $\mathcal{L}_{CL}$ and $\mathcal{L}_{LM}$:

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

3.5. Inference

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

Documents Retrieval

For this task, we can use the bi-encoder part of GeAR to compute the similarity between query and document like the previous classic retrieval method, without introducing any additional parameters and computation cost.

Fine-Grained Units Localization

The fusion encoder in GeAR calculates the fusion embedding of query and document through cross attention. We use the cross attention weights of the query on the tokens in the document to locate the units that the query pays the most attention to in the document.

Information Generation

For this task, we feed the fusion embedding to the text decoder and enable autoregressive decoding. In GeAR, information generation is actually an auxiliary task, and we will present the generative performance of the model in experiments, both in terms of quantitative metrics and qualitative analysis.

4. Experiments

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

4.1. Setup

Datasets

For Question Answer Retrieval, we sampled 30M data from PAQ[38] datasets to train GeAR, and sampled 1M documents and 20k queries as test set. We also evaluate the performance on another 3 QA datasets: SQuAD[39], NQ[40], and TriviaQA[41]. These test datasets are all held out to observe the generalizability of compared methods. 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

To better observe the effectiveness of GeAR, we use "BERT-base-uncased"[17] to initialize the encoders in GeAR. We trained the model for 10 epochs using a batch size of 48 (QAR) / 16 (RIR) on 16 AMD MI200 GPUs with 64GB memory. We use the AdamW[42] optimizer with a weight decay of 0.05. The full hyperparameters and training settings are detailed in Appendix C.

Baselines

We compare our approach to two classes of baseline methods, one class of text representation models that have been adequately trained on a large corpus, including SBERT[18], specifically "all-mpnet-base"[43], E5[20], BGE[44], and GTE[21]. We use both base-level models for this comparison. The other category consists different training pipelines that unify the training data and starting points, including SBERT[18] and BGE[44]. We retrained them all using the "bert-base-uncased" to initialize and aligned the training data, referred to as SBERT_{RT} and BGE_{RT} in the following.

SQuAD		**NQ**		**TriviaQA**		**PAQ**		**RIR**	
EM	F1	EM	F1	EM	F1	EM	F1	Rouge-1	Rouge-L
61.2	65.3	66.1	61.0	47.4	60.0	88.1	92.4	87.4	87.1

Table 3. Generation performance of GeAR on different tasks.

4.2. Overall performance

In this section, we present the overall performance on Documents Retrieval, Units Localization, and Information Generation.

Documents Retrieval

Firstly, we report the comparison with existing methods on documents retrieval task in Table 1. The results demonstrate that GeAR delivers competitive performance across multiple datasets, even with limited training data. As a reference, the pre-trained SBERT model used 1.17B sentence pairs, with partial overlap between its training data and our evaluation data. To ensure a fair comparison, we retrained SBERT[2] and BGE[3] using their open source training pipelines, aligned training data and initialization settings. As shown in the retrained model section in Table 1, GeAR achieves superior performance, underscoring the effectiveness of our training approach.

Units Localization

Next, we evaluate the performance of each method on the units localization task. In the evaluation process, we provide the query and the document (q, d) to the model and observe whether it is able to find 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, GeARlocates units based on the cross attention weights for each sentence given the document and the query, as described in Section 3.5. The results are reported in Table 2. We found that GeAR came out ahead on all metrics, and that GeAR did not require further chunking and encoding of the document. It is observed that SBERT_{RT} and BGE_{RT} exhibit suboptimal performance, as their training objective focus solely on optimizing the overall semantic similarity between the document and the query, neglecting the fine-grained semantic relationships. In contrast, GeAR benefits from the joint end-to-end training of retrieval and generation tasks, enabling it not only to retrieve documents closely aligned with the query but also to effectively attend to fine-grained information within the document.

	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	0.991
BGE$_{RT}$	0.455	0.538	0.601	0.656	0.288	0.475	0.409	0.466	0.897	0.888
GeAR	**0.810**	**0.874**	**0.765**	**0.871**	**0.515**	**0.808**	**0.885**	**0.965**	**0.954**	**0.897**

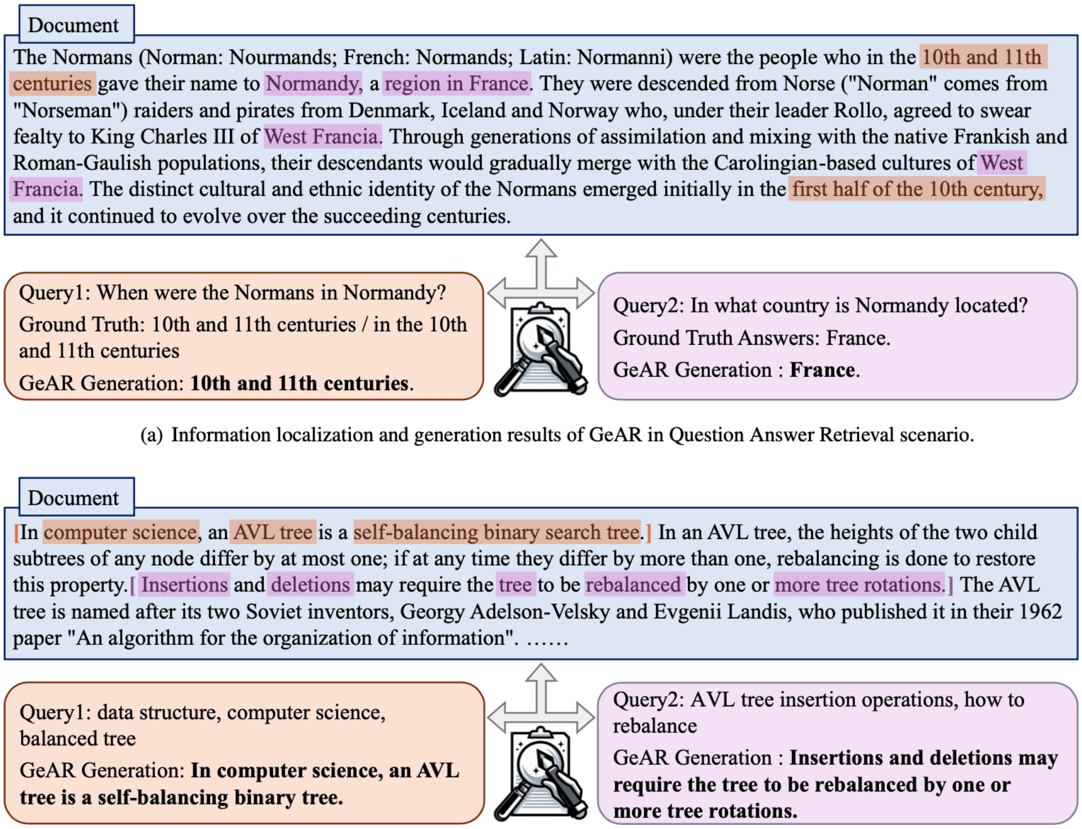
Table 2. Comparison of units localization performance on different datasets, where R@k stands for Recall@k, M@k stands for MAP@k.

Information Generation

Although generation serves only as an auxiliary task in GeAR, we are nonetheless interested in evaluating its generation performance. Table 3 reports the Exact Match (EM) and F1 scores on the QA datasets, and the Rouge[45] scores on the RIR dataset. Notably, GeAR achieves strong performance on test sets with distributions similar to the training data, such as PAQ and RIR, and performs reasonably well on other test sets. Additionally, Figure 3 illustrates examples of GeAR's ability to generate correct answers and relevant information, demonstrating its satisfactory generation capabilities.

	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	**0.883**	**0.762**	**0.747**	**0.567**	**0.660**	**0.398**	**0.940**	**0.855**	**0.961**	**0.903**
GeAR$_{w/o\mathcal{L}_{LM}}$	**0.889**	**0.776**	**0.755**	**0.565**	**0.660**	**0.399**	**0.955**	**0.877**	**0.963**	**0.907**

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



Document

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse ("Norman" comes from "Norseman") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

Query1: When were the Normans in Normandy?

Ground Truth: 10th and 11th centuries / in the 10th and 11th centuries

GeAR Generation: **10th and 11th centuries**.

Query2: In what country is Normandy located?

Ground Truth Answers: France.

GeAR Generation : **France**.

(a) Information localization and generation results of GeAR in Question Answer Retrieval scenario.

Document

[In computer science, an AVL tree is a self-balancing binary search tree.] In an AVL tree, the heights of the two child subtrees of any node differ by at most one; if at any time they differ by more than one, rebalancing is done to restore this property.[Insertions and deletions may require the tree to be rebalanced by one or more tree rotations.] The AVL tree is named after its two Soviet inventors, Georgy Adelson-Velsky and Evgenii Landis, who published it in their 1962 paper "An algorithm for the organization of information". ……

Query1: data structure, computer science, balanced tree

GeAR Generation: **In computer science, an AVL tree is a self-balancing binary tree.**

Query2: AVL tree insertion operations, how to rebalance

GeAR Generation : **Insertions and deletions may require the tree to be rebalanced by one or more tree rotations.**

(b) Information localization 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 information localization of GeAR. In the two scenarios of Question Answer retrieval and Related Information Retrieval, we propose two different queries for one document and 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. We also show the generated results of GeAR.

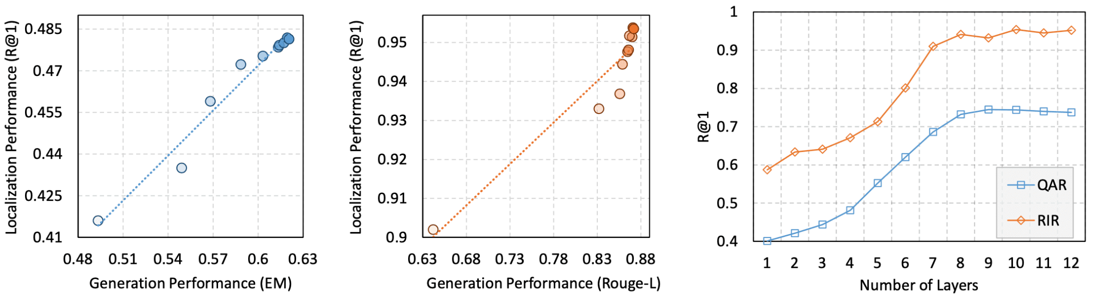
4.3. Analysis

Visualization of Information Localization

Figure 3 illustrates the information localization and generation results of GeAR across different scenarios. We provide two distinct queries for one document and highlight the top 10 tokens with the highest cross attnetion weights corresponding to each queries. In Figure 3(a), the two queries focus on time and location, respectively. GeAR not only gave the correct answers to both queries but also dynamically adjusts its query-specific focus: it assigns higher attention weights to time-related tokens in response to the first query and prioritizes tokens related to countries and regions in response to the second query. In Figure 3(b), GeAR will focus on the definition of the AVL tree itself, and the insertion, deletion, rotation and rebalancing of the AVL tree, and generate corresponding sentences. It can be seen that the added generation task has brought improvements to the model in terms of performance and qualitative effects, making it accurate in localization and generation.

Correlation of Generation and Localization

In this section, we analyze the relationship between the generation and localization tasks. As illustrated in Figure 4(a) and 4(b), we plot the performance coordinates from epoch 1 to epoch 10 during training, where the horizontal axis represents the generation performance and the vertical axis represents the localization performance. The results reveal a strong correlation between the two tasks. This observation demonstrates that the generation task, designed as a proxy, effectively enhances the model's ability to extract fine-grained information from documents. These findings highlight the synergistic relationship between generation and localization.



(a) On QAR tasks.

(b) On RIR tasks.

(c) Localization performance of different layers.

Figure 4. Plots of generation and localization performance on (a) QAR tasks and (b) RIR tasks as training progresses. (c) shown the localization performance at different layers.

Localization performance of different layers

In GeAR, the fusion encoder and decoder interact through the cross attention module at each layer. To investigate the relationship between localization performance and model depth, we plot the localization performance using cross attention weights across different layers in Figure 4(c). The results indicate that high-level token embeddings perform well, as they capture rich semantic information through deeper layers of the network. Interestingly, we observe that the highest layer

doi.org/10.32388/GWQ6LR

does not yield the best localization performance. Instead, peak performance is achieved in the last 3 to 4 layers[4]. This phenomenon may arise because the representations in the highest layer are optimized to serve the final task rather than intermediate localization. Similar findings have been reported in prior studies involving encoder-only and decoder-only models[46][47].

The Affect of Language Modeling Objectives

In this work, we utilize only the information corresponding to the query as supervision and incorporate a language modeling objective. It enables the model to achieve stronger capabilities in both information localization and generation, without requiring additional loss functions or complex module designs. However, as a trade-off, we observe a slight decrease in retrieval performance when compared to using only the contrastive learning objective for the retrieval task, as shown in the last two rows of Table 1. How to further design the balance between the two training objectives from the perspective of multi-task learning so that they benefit from each other is a point that can still be explored in the future.

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: **G**eneration **A**ugmented **R**etrieval (**GeAR**). GeAR enhances fine-grained information localization and generation capabilities by incorporating a decoder and a lightweight cross-attention layer, while maintaining the efficiency of a bi-encoder. Experimental results across multiple retrieval tasks and two different scenarios demonstrate that GeAR achieves competitive performance. Furthermore, its ability to accurately and reasonably localize information makes it particularly promising for downstream tasks such as web search, semantic understanding, and retrieval-augmented generation (RAG). We hope this work offers valuable insights into the gradual unification of natural language understanding and generation paradigms, paving the way for more versatile 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 [48], 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.

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.

Appendix 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[36]. 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[49] and "Llama-3.1-70B-Instruct" [50] 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[44] 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 found 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.

Appendix 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[38] 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 5.

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 4. Hyperparameter settings

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

Table 5. Training data statistics.

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 6. The evaluation data statistics for the 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 7. The evaluation data statistics for the units localization and information generation tasks.

B.2. Evaluation

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

Documents Retrieval

First, for the 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[39], NQ[40], TriviaQA[41], and RIR 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 6.

Units Localization and Information Generation

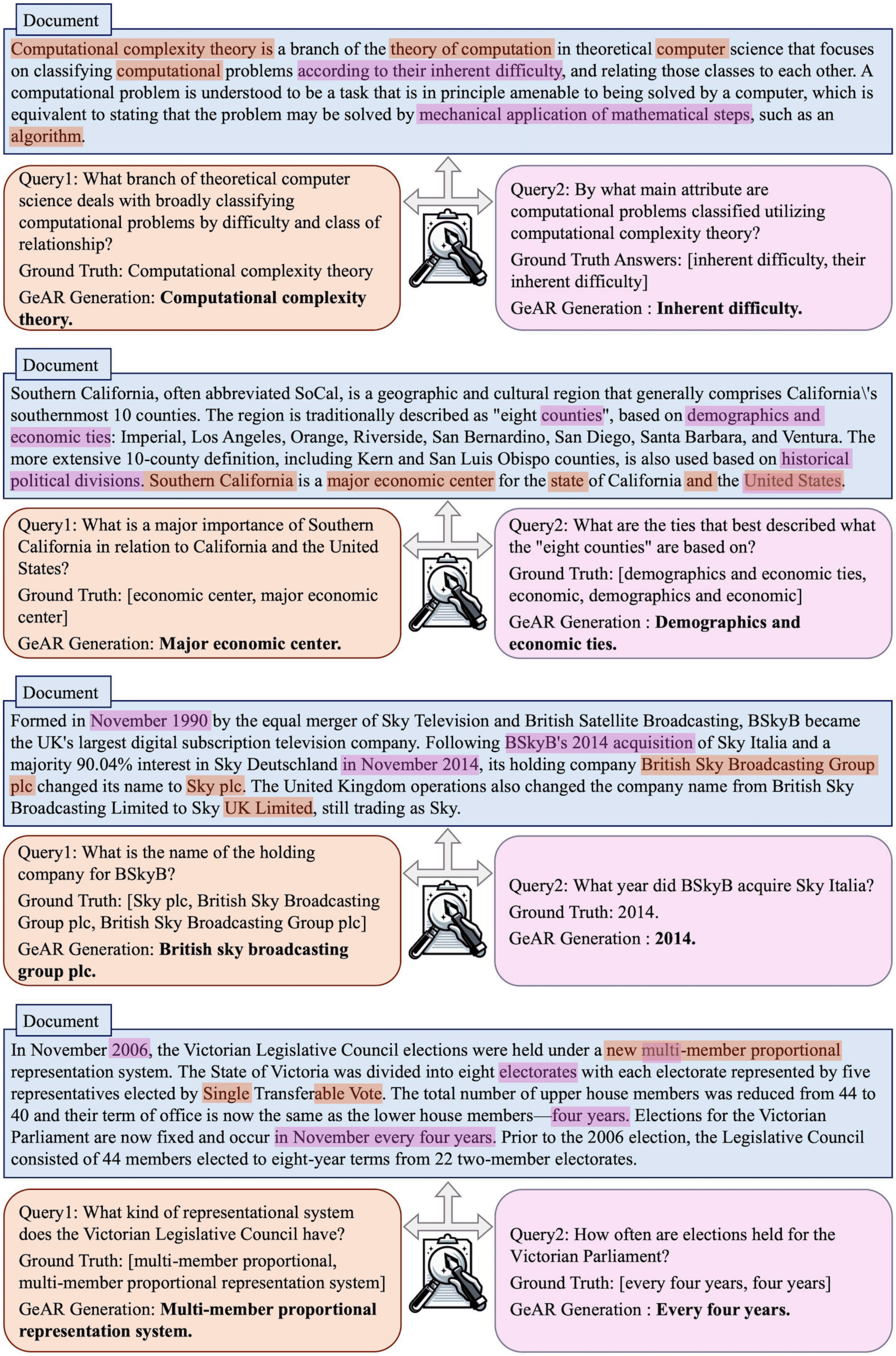
For these two tasks, we directly use the test set data corresponding to the respective datasets. Therefore, their number is consistent with the number of queries in Table 6. For the RIR dataset, we sample 10k records as the test set. The evaluation data statistics for the units localization and information generation tasks are shown in Table 7.

Appendix 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 4. We use FAISS[51][52] to store and search for vectors. The 2 encoders and 1 decoders in GeAR are the same size as "bert-base"[17], 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.

Appendix 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.

Footnotes

[1] Note that in the generation task of the QAR scenario, the ground truth is the answer itself, not the sentence u. But in the RIR scenario and the localization task, we all used the sentence u.

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

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

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

References

1. [a, b]*Chen D, Fisch A, Weston J, Bordes A (2017). "Reading Wikipedia to answer open-domain questions". 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. doi:10.18653/v1/P17-1171.*
2. [a, b]*Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, Chen D, Yih W (2020). "Dense passage retrieval for open-domain question answering". Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pages 6769–6781. doi:10.18653/v1/2020.emnlp-main.550.*
3. ^*Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, Küttler H, Lewis M, Yih W, Rocktäschel T, Riedel S, Kiela D (2020). "Retrieval-augmented generation for knowledge-intensive nlp tasks". Advances in Neural Information Processing Systems. 33: 9459–9474.*
4. ^*Liu H, Liu J, Huang S, Zhan Y, Sun H, Deng W, Wei F, Zhang Q (2024). "se^2: Sequential example selection for in-context learning". Findings of the Association for Computational Linguistics ACL 2024. pages 5262–5284. https://aclanthology.org/2024.findings-acl.312.*
5. ^*Gao Y, Xiong Y, Gao X, Jia K, Pan J, Bi Y, Dai Y, Sun J, Wang M, Wang H (2024). "Retrieval-augmented generation for large language models: A survey."*
6. ^*Brito E, Iser H (2023). "Maxsime: Explaining transformer-based semantic similarity via contextualized best matching token pairs". 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. doi:10.1145/3539618.3592017.

7. ^*Luo K, Liu Z, Xiao S, Liu K (2024). "Bge landmark embedding: A chunking-free embedding method for retrieval augmented long-context large language models". arXiv preprint arXiv:2402.11573. arXiv:2402.11573.*
8. ^*Günther M, Mohr I, Williams DJ, Wang B, Xiao H (2024). "Late chunking: contextual chunk embeddings using long-context embedding models". arXiv preprint arXiv:2409.04701. Available from: https://arxiv.org/abs/2409.04701.*
9. a, b*Liu NF, Lin K, Hewitt J, Paranjape A, Bevilacqua M, Petroni F, Liang P (2024). "Lost in the middle: How language models use long contexts". Transactions of the Association for Computational Linguistics. 12: 157–173. doi:10.1162/tacl_a_00638.*
10. a, b*An S, Ma Z, Lin Z, Zheng N, Lou JG (2024). "Make your llm fully utilize the context". In NeurIPS 2024.*
11. a, b*Wang M, Chen L, Cheng F, Liao S, Zhang X, Wu B, Yu H, Xu N, Zhang L, Luo R, Li Y, Yang M, Huang F, Li Y (2024). "Leave no document behind: Benchmarking long-context LLMs with extended multi-doc QA". Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 5627–5646. Miami, Florida, USA: Association for Computational Linguistics.*
12. a, b*Gao T, Yen H, Yu J, Chen D (2023). "Enabling large language models to generate text with citations". Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pages 6465–6488, Singapore. Association for Computational Linguistics. doi:10.18653/v1/2023.emnlp-main.398.*
13. a, b*Zhang J, Bai Y, Lv X, Gu W, Liu D, Zou M, Cao S, Hou L, Dong Y, Feng L, et al. (2024). "Longcite: Enabling llms to generate fine-grained citations in long-context qa". arXiv preprint arXiv:2409.02897.*
14. ^*Khattab O, Zaharia M (2020). "Colbert: Efficient and effective passage search via contextualized late interaction over bert". Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. pp. 39–48.*
15. ^*Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013). "Distributed representations of words and phrases and their compositionality". Advances in Neural Information Processing Systems. 26.*
16. ^*Pennington J, Socher R, Manning CD (2014). "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543.*

17. ^a, b, c^ *Devlin J, Chang MW, Lee K, Toutanova K (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." 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). Minneapolis, Minnesota: Association for Computational Linguistics. pp. 4171–4186. doi:10.18653/v1/N19-1423. Available from: https://aclanthology.org/N19-1423.*

18. ^a, b, c^ *Reimers N, Gurevych I (2019). "Sentence-BERT: Sentence embeddings using Siamese BERT-networks". Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. doi:10.18653/v1/D19-1410.*

19. ^ *Huang PS, He X, Gao J, Deng L, Acero A, Heck L (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.*

20. ^a, b^ *Wang L, Yang N, Huang X, Jiao B, Yang L, Jiang D, Majumder R, Wei F (2022). "Text embeddings by weakly-supervised contrastive pre-training". arXiv preprint arXiv:2212.03533. Available from: https://arxiv.org/abs/2212.03533.*

21. ^a, b^ *Li Z, Zhang X, Zhang Y, Long D, Xie P, Zhang M (2023). "Towards general text embeddings with multi-stage contrastive learning". arXiv preprint arXiv:2308.03281. Available from: https://arxiv.org/abs/2308.03281.*

22. ^ *Gao T, Yao X, Chen D (2021). "SimCSE: Simple contrastive learning of sentence embeddings". Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pages 6894–6910. doi:10.18653/v1/2021.emnlp-main.552.*

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

24. ^ *Hofstätter S, Lin SC, Yang JH, Lin J, Hanbury A (2021). "Efficiently teaching an effective dense retriever with balanced topic aware sampling". Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 113–122.*

25. ^ *Muennighoff N, Su H, Wang L, Yang N, Wei F, Yu T, Singh A, Kiela D (2024). "Generative representational instruction tuning". http://arxiv.org/abs/2402.09906.*

26. ^ *Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, et al. (2021). "Learning transferable visual models from natural language supervision." In International conf*

erence on machine learning, pages 8748–8763. PMLR.

27. a, bLi J, Selvaraju R, Gotmare A, Joty S, Xiong C, Hoi SCH (2021). "Align before fuse: Vision and language representation learning with momentum distillation". Advances in Neural Information Processing Systems. 34: 9694–9705.

28. a, b, cLi J, Li D, Xiong C, Hoi S (2022). "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation." In International Conference on Machine Learning, pages 12888–12900. PMLR.

29. ^Seo M (2016). "Bidirectional attention flow for machine comprehension". arXiv preprint arXiv:1611.01603. Available from: arXiv:1611.01603.

30. ^Wang S (2016). "Machine comprehension using match-lstm and answer pointer". arXiv preprint arXiv:1608.07905. Available from: https://arxiv.org/abs/1608.07905.

31. ^Hu Xu, Liu B, Shu L, Yu P (2019). "BERT post-training for review reading comprehension and aspect-based sentiment analysis". 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). Minneapolis, Minnesota: Association for Computational Linguistics. pp. 2324–2335. doi:10.18653/v1/N19-1242.

32. ^Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (2016). "Hierarchical attention networks for document classification". Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pages 1480–1489. doi:10.18653/v1/N16-1174.

33. ^Liu Y, Hashimoto K, Zhou Y, Yavuz S, Xiong C, Yu P (2021). "Dense hierarchical retrieval for open-domain question answering". Findings of the Association for Computational Linguistics: EMNLP 2021. Punta Cana, Dominican Republic: Association for Computational Linguistics. pp. 188–200. doi:10.18653/v1/2021.findings-emnlp.19.

34. ^Arivazhagan MG, Liu L, Qi P, Chen X, Wang WY, Huang Z (2023). "Hybrid hierarchical retrieval for open-domain question answering". Findings of the Association for Computational Linguistics: ACL 2023. 10680–10689. doi:10.18653/v1/2023.findings-acl.679.

35. ^Chen HY, Lai Z, Zhang H, Wang X, Eichner M, You K, Cao M, Zhang B, Yang Y, Gan Z (2024). "Contrastive localized language-image pre-training". arXiv preprint arXiv:2410.02746. Available from: https://arxiv.org/abs/2410.02746.

36. a, bWikimedia Foundation. "Wikimedia downloads."

37. a, b He K, Fan H, Wu Y, Xie S, Girshick R (2020). "Momentum contrast for unsupervised visual representation learning." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

38. a, b Lewis P, Wu Y, Liu L, Minervini P, Küttler H, Piktus A, Stenetorp P, Riedel S (2021). "PAQ: 65 Million Probably-Asked Questions and What You Can Do With Them". Transactions of the Association for Computational Linguistics. 9: 1098--1115. doi:10.1162/tacl_a_00415.

39. a, b Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016). "SQuAD: 100,000+ Questions for Machine Comprehension of Text". In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas: Association for Computational Linguistics; p. 2383-2392. doi:10.18653/v1/D16-1264. Available from: https://aclanthology.org/D16-1264.

40. a, b Kwiatkowski T, Palomaki J, Redfield O, Collins M, Parikh A, Alberti C, Epstein D, Polosukhin I, Devlin J, Lee K, Toutanova K, Jones L, Kelcey M, Chang MW, Dai AM, Uszkoreit J, Le Q, Petrov S (2019). "Natural questions: A benchmark for question answering research". Transactions of the Association for Computational Linguistics. 7: 452–466. doi:10.1162/tacl_a_00276.

41. a, b Joshi M, Choi E, Weld D, Zettlemoyer L (2017). "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics. pp. 1601--1611. doi:10.18653/v1/P17-1147. Available from: https://aclanthology.org/P17-1147.

42. ^ Loshchilov I (2017). "Decoupled weight decay regularization". arXiv preprint arXiv:1711.05101. Available from: https://arxiv.org/abs/1711.05101.

43. ^ Song K, Tan X, Qin T, Lu J, Liu T (2020). "Mpnet: Masked and permuted pre-training for language understanding". Advances in Neural Information Processing Systems. 33: 16857–16867. Link to paper.

44. a, b, c Xiao S, Liu Z, Zhang P, Muennighoff N, Lian D, Nie JY (2024). "C-pack: Packed resources for general chinese embeddings". 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. doi:10.1145/3626772.3657878.

45. ^ Lin CY (2004). "ROUGE: A Package for Automatic Evaluation of Summaries". In: Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics; p. 74-81.

46. ^ Jawahar G, Sagot B, Seddah D (2019). "What does BERT learn about the structure of language?" In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3651–365

7, Florence, Italy. Association for Computational Linguistics. doi:10.18653/v1/P19-1356.

47. ^*Skean O, Arefin MR, LeCun Y, Shwartz-Ziv R (2024). "Does representation matter? exploring intermediate layers in large language models". arXiv preprint arXiv:2412.09563. Available from: https://arxiv.org/abs/2412.09563.*
48. ^*Zhu D, Wang L, Yang N, Song Y, Wu W, Wei F, Li S (2024). "LongEmbed: Extending embedding models for long context retrieval". Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. Miami, Florida, USA: Association for Computational Linguistics. pp. 802–816. doi:10.18653/v1/2024.emnlp-main.47.*
49. ^*Kwon W, Li Z, Zhuang S, Sheng Y, Zheng L, Yu CH, Gonzalez JE, Zhang H, Stoica I (2023). "Efficient memory management for large language model serving with pagedattention". Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.*
50. ^*Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, Mathur A, Schelten A, Yang A, Fan A, et al. (2024). "The llama 3 herd of models". arXiv preprint arXiv:2407.21783. Available from: https://arxiv.org/abs/2407.21783.*
51. ^*Douze M, Guzhva A, Deng C, Johnson J, Szilvasy G, Mazaré PE, Lomeli M, Hosseini L, Jégou H (2024). "The faiss library". http://arxiv.org/abs/2401.08281.*
52. ^*Johnson J, Douze M, Jégou H (2019). "Billion-scale similarity search with GPUs". IEEE Transactions on Big Data.* 7 *(3): 535–547.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.