LLM-Augmented Retrieval: Generalize Retriever Models to Specific Domains Without Finetuning

Anonymous ACL submission

Abstract

Recent advancements in embedding-based retrieval, commonly referred to as dense retrieval, have achieved state-of-the-art results, surpassing the performance of traditional sparse or bag-of-words methodologies. Embedding-based techniques are extensively utilized in enterprise and domain-specific search applications, which often require finetuning on domain-specific data to enhance retrieval performance. However, the scarcity of domain-specific data and the complexity of finetuning present significant challenges in developing efficient domain-specific retrieval systems. This paper introduces a training-free, model-agnostic document-level embedding framework augmented by a large language model (LLM). This framework significantly enhances the efficacy of prevalent retriever models, including Bi-encoders (such as Contriever and DRAGON) and late-interaction models (such as ColBERTv2), and generalizes them into new domains. As a result, this approach has achieved state-of-the-art performance on benchmark datasets like LoTTE and BEIR, highlighting its potential to advance information retrieval processes, particularly in domain-specific contexts.

1 Introduction

In the realm of information retrieval (IR), the pursuit of more precise and efficient retrieving methods has been a continuous endeavor. Traditional IR systems have predominantly relied on sparse techniques, such as the bag-of-words (HaCohen-Kerner et al., 2020; Robertson et al., 1995; Zhang et al., 2010), but often fail to capture the semantic richness of queries and documents due to their dependence on exact keyword matches. Embedding-based retrieval (Huang et al., 2020), also known as dense retrieval, offers improved retrieval performance by converting text into dense vector spaces, where semantically similar texts are positioned in close proximity, thereby enabling the capture of deep semantic relationships that are not readily discernible through keyword matching alone.

In addition to innovations in model architectures, techniques such as query rewriting (Gottlob et al., 2014; He et al., 2016; Singh and Sharan, 2017; Xiong and Callan, 2015) have proven effective in enhancing query information from the user's perspective before conversion into dense vectors. Conversely, we propose that enriching document embeddings can also significantly improve text retrieval quality. Importantly, this process can be conducted offline in advance, thereby reducing the time required for online inference. In the past, scalable methods for augmenting document-related information have been challenging to implement; however, the emergence of large language models (LLMs) provides a viable solution. This paper introduces a document enrichment framework designed to enhance retrieval performance by leveraging language models without the need for finetuning.

Our primary contributions are as follows: 1) We present LLM-augmented retrieval, a training-free, model-agnostic framework 1 that enhances contextual information within the vector embeddings of documents, thereby improving the performance of existing retrievers across various domains; 2) We validate this framework across a range of models and extensive datasets, achieving state-of-the-art performance improvements over the original models without any finetuning; 3) Our framework exhibits strong generalizability to new domains, facilitating its adoption in domain-specific retrieval applications or enterprise search scenarios.

2 LLM-augmented Retrieval Framework

2.1 Synthetic Relevant Queries

The concept of synthetic relevant queries arises from a reevaluation of the traditional reliance on

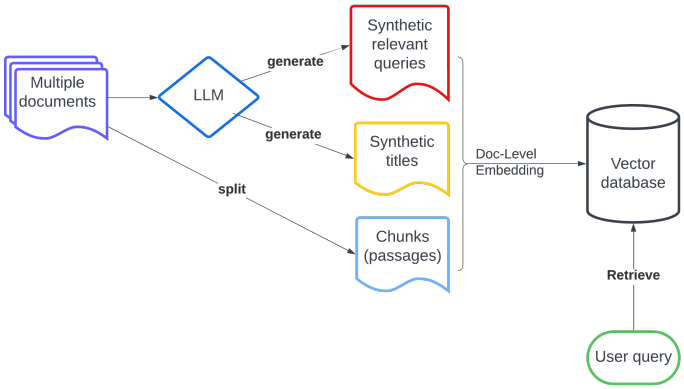


Figure 1: Overall view on LLM-augmented retrieval framework. Synthetic relevant queries and synthetic titles are generated from LLM and then assembled into doc-level embedding together with chunks (passages) split from the original document. The final retrieval is based on the similarity between user query and the doc-level embedding.

similarity metrics for determining relevance in retrieval tasks (Jones and Furnas, 1987). These metrics, often based on the dot product or cosine similarity of encoded vectors, may not adequately capture the semantic nuances crucial for relevance. For example, the queries "Who is the first president of the United States?" and "Who became the first president of America?" might yield high similarity scores but differ in semantic relevance. The desired document, such as a biography of George Washington, might not score highly against these queries. However, if synthetic queries generated from Washington's biography include "Who became the first president of America?", it becomes possible to bridge the semantic gap. The synthetic query not only reflects the document's content from various perspectives but also enhances the matching process with relevant user queries, as illustrated in Figure 2a. Through synthetic relevant queries, the relevance relationship is not solely expressed by the similarity between user queries and documents but also inferred from the similarity between user queries and pre-stored relevant queries.

2.2 Title

A document's title is a critical determinant of its relevance and utility in response to user search queries. As the primary element encountered in search results, titles significantly influence user decision-making regarding link selection. Effective titles provide essential context and keywords, enabling users to rapidly assess content and objectives. When original documents possess titles, they can be leveraged to enhance search relevance. Conversely, for untitled documents, large language models can generate synthetic titles that capture the essence and main themes, thereby aligning the document with user informational needs. Whether derived directly or synthesized through advanced modeling, titles play a crucial role in optimizing the search and discovery process.

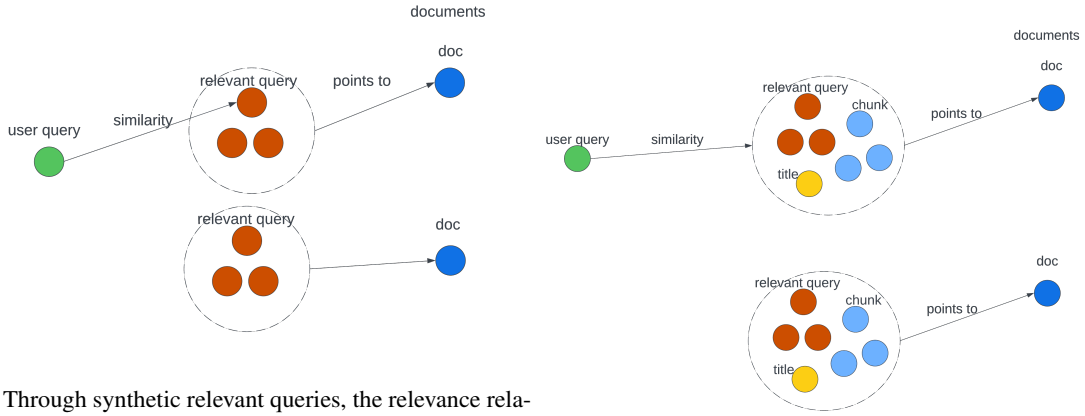
2.3 Document chunks

Document chunking is a methodological approach that involves segmenting large documents into smaller, manageable units (chunks or passages) to facilitate analysis and processing (Chen et al., 2023; Finardi et al., 2024; Lewis et al., 2020). This process groups related information segments within the constraints of retrieval models' context windows, which limit input length. Chunks are derived directly from original documents without language model augmentation.

In practice, lengthy documents are divided into chunks containing tokens within the model's context window limit. Optimal chunk size varies across Bi-encoder retrieval models, whereas token-level late-interaction models like ColBERT or ColBERTv2 do not require chunking due to their token-level similarity score calculations. This distinction highlights the importance of model-specific considerations when implementing chunking strategies in information retrieval systems.

2.4 Doc-level embedding

For clarity, we refer to the information sources—synthetic relevant queries, titles, and chunks—as the fields of a document. These fields represent the semantics of the original document from various perspectives and are



(a) Through synthetic relevant queries, the relevance relationship is not solely expressed by the similarity between user queries and documents but also inferred from the similarity between user queries and pre-stored relevant queries.

(b) The graphic representation of "relevance" in doc-level embedding

Figure 2: Graphic representation of synthetic queries, titles, passage chunks in doc-level embedding

integrated into the document-level embedding (see Figure 2b). This embedding is static, allowing it to be pre-computed and cached for efficient retrieval. Indexes of these embeddings can be pre-built to expedite the retrieval process, with each embedding linking back to the original document.

The Bi-encoder architecture (Cer et al., 2018; Karpukhin et al., 2020) is a widely used approach in dense retrieval, consisting of two encoders (shared or distinct) that generate vector representations for user queries and documents. The relevance between queries and documents is determined by computing the similarity between these vectors. To augment document embeddings with synthetic relevant queries, titles and document chunks, we propose a modified similarity computation:

Definition 2.1. Similarity score for query-document pairs in Bi-encoders:

$$sim(q,d) = \max_i s(q,c_i) + s(q,d) \quad (1)$$

where

$$s(q,d) = s(q, \frac{w_c}{m}\sum_i^m c_i + \frac{w_{q^*}}{n}\sum_j^n q_j^* + w_{t^*}t^*) \quad (2)$$

The term $\max_i s(q,c_i)$ computes the traditional maximum similarity score across query-chunk embedding pairs, where s denotes the similarity function, q represents the search query's embedding vector, and c_i is the embedding vector for the i-th document chunk. This approach is prevalent in modern embedding-based retrieval systems, focusing on the similarity between a query and the most relevant document chunk. The second term $s(q,d)$ introduces a novel aspect by incorporating additional augmented information at the document level. Here, c are the chunk embedding vectors mentioned above, $q*$ are the embedding vectors of synthetic relevant queries, t^* is the title embedding vector, while w_c, w_q^*, w_{t^*} are the corresponding document field weights. (Arora et al., 2017) also suggests averaging these vectors to represent the entire document, as an approach we adopt for both chunk and synthetic query fields. This method has proven effective in our experiments, though more sophisticated techniques could be explored in future work.

Given that the similarity function is linear[1], the equation can be transformed to:

$$sim(q,d) = \max_i s(q, c_i + \frac{w_c}{m}\sum_i^m c_i + \frac{w_{q^*}}{n}\sum_j^n q_j^* + w_{t^*}t^*) \quad (3)$$

This simplification allows us to treat $c_i + \frac{w_c}{m}\sum_i^m c_i + \frac{w_{q^*}}{n}\sum_j^n q_j^* + w_{t^*}t^*$ as the composite embedding vector for each document chunk c_i, enabling the use of algorithms like approximate nearest neighbors (Indyk and Motwani, 1998) for efficient document retrieval.

In token-level late-interaction models like ColBERT and ColBERTv2, user queries and documents are encoded into token-level vector representations independently. The "late interaction" involves computing cosine similarity or dot product scores between these representations at the token level. To incorporate augmented queries and

[1]Both dot product and cosine similarity are linear when embedding vectors are normalized to unit length.

titles, we append them to the original documents, enabling the model to utilize these additional signals in its similarity calculation.

3 Experiments

3.1 Datasets and Models

BEIR Data The BEIR (Benchmark for Evaluating Information Retrieval) dataset (Thakur et al., 2021) serves as a comprehensive benchmark for assessing various information retrieval (IR) models, particularly in out-of-domain scenarios. Designed to overcome the limitations of previous datasets, BEIR offers a diverse and extensive collection of queries and passages across a broad range of topics. This diversity enables a more thorough and robust evaluation of IR models.

LoTTE Data The LoTTE dataset (Santhanam et al., 2021) is specifically crafted for Long-Tail Topic-stratified Evaluation, focusing on natural user queries linked to long-tail topics that are often underrepresented in entity-centric knowledge bases like Wikipedia.

Contriever The Contriever model employs the Roberta-base (Liu et al., 2019) architecture, trained on Wiki passages (Karpukhin et al., 2020) and CC100 (Conneau et al., 2019) data through contrastive learning. It features 125 million parameters, a context window of 512 tokens, 12 layers, 768 hidden dimensions, and 12 attention heads. In this model, a single Roberta-base model serves as both the query encoder and context encoder, following a shared "Two Tower" Bi-encoder architecture.

DRAGON Similarly, the DRAGON model utilizes the Roberta-base architecture. However, unlike Contriever, DRAGON employs separate Roberta-base models for the query encoder and context encoder. This model's checkpoint was trained and released publicly by the author.

ColBERTv2 For ColBERTv2, the bert-base-uncased model architecture is adopted, consistent with the default settings in the original paper. This model comprises 110 million parameters and a context window of 256 tokens, with 12 layers, 768 hidden dimensions, and 12 attention heads. The checkpoint for ColBERTv2 was trained on the MS-MARCO dataset (Nguyen et al., 2016) and provided by the author.

3.2 Implementation Details

We choose open source Llama3-8B (Dubey et al., 2024; Touvron et al., 2023a,b) for both synthetic queries generation and titles generation. The prompt templates are in Table 7 and 8.

For Bi-encoders, we implemented the doc-level embeddings as mentioned above with chunk_size=64 and chose w_{q^*}=1.0, w_{t^*}=0.5, w_c=0.1 for the Contriever model and w_{q^*}=0.6, w_{t^*}=0.3, w_c=0.3 for the DRAGON model. Those hyperparameters are selected based on the dev set of BEIR-ArguAna and then fixed across all the evaluation sets. The hyperparameters seem to generalize well. For ColBERTv2, as mentioned previously, we concatenate the title with all the synthetic queries for each document and make it an additional "passage" of the original document. We set index_bits=8 when building the ColBERT index. All the results in the below sections are from single runs.

3.3 Results

We evaluate the performance of vanilla Bi-encoder models and our proposed LLM-augmented method on the BEIR dataset, reporting nDCG@10 in Table 1. The results demonstrate that integrating LLM-augmented retrieval with document-level embeddings significantly improves nDCG@10 metrics for Contriever and DRAGON models. Notably, the improvement is more pronounced for Contriever, a weaker retriever model compared to DRAGON.

On the LoTTE dataset, we compare all three models in both vanilla and LLM-augmented modes, reporting their Recall@3 (R@3) in Table 2. The results show noticeable improvements across all models, with the magnitude of enhancement being more significant when the base retriever model is weaker.

3.4 LLM Augmentation Analysis

Table 3 gives an overview on the number of documents per dataset (N_D, in thousands), the number of total tokens in documents (N_{T_D}, in thousands), the average number of tokens per document (N_{T_D}/N_D), the number of synthetic queries generated (N_{q^*}, in thousands), the total number of total synthetic query tokens generated ($N_{T_{q^*}}$, in thousands), the average number of synthetic query per document (N_{q^*}/N_D), the average number of synthetic query tokens per document ($N_{T_{q^*}}/N_D$) and the average number of synthetic query tokens per synthetic query ($N_{T_{q^*}}/N_{q^*}$). On average 6 synthetic relevant queries are generated per document and the token count in the generated synthetic queries is comparable to the token count

nDCG@10	Contriver		Dragon	
BEIR	Vanilla	LLM-Aug	Vanilla	LLM-Aug
ArguAna	33.2	34.8	52.0	50.3
FiQA	3.0	28.7	8.0	42.5
Quora	83.1	83.3	89.0	88.3
SCIDOCS	17.0	24.6	30.4	31.8
SciFact	53.1	58.0	67.1	68.0
Climate-FEVER	7.3	27.8	32.0	37.3
MS MARCO	63.1	72.1	99.6	99.2
DBPedia	45.0	57.7	84.6	83.1
Touche-2020	51.1	69.9	72.4	78.0
NFCorpus	32.3	48.1	49.8	50.6
Trec-COVID	50.8	82.9	95.5	95.0
CQADupStack	16.4	29.4	35.8	40.0
FEVER	8.2	52.7	74.9	77.7
HotpotQA	45.8	58.6	81.4	78.6
NFCorpus	32.3	48.1	54.5	57.3

Table 1: The performance of vanilla retriever models vs LLM-augmented retriever performance on BEIR datasets.

R@3		Contriver		Dragon		ColbertV2	
LoTTE		Vanilla	LLM-Aug	Vanilla	LLM-Aug	Vanilla	LLM-Aug
Lifestyle	Search	33.6	60.2	56.0	76.3	79.3	80.0
	Forum	43.7	62.4	52.7	68.8	69.9	73.1
Recreation	Search	19.5	46.1	42.5	64.7	66.8	71.0
	Forum	34.9	54.6	45.6	60.8	63.4	67.5
Science	Search	10.1	29.0	26.0	45.0	50.7	50.2
	Forum	10.5	24.0	25.8	31.0	39.3	40.3
Technology	Search	12.4	35.6	35.9	52.9	59.4	59.6
	Forum	18.3	36.6	28.5	41.9	45.0	46.3
Writing	Search	27.5	57.2	58.0	70.3	74.2	75.4
	Forum	39.5	59.7	53.0	65.2	69.6	71.5

Table 2: The performance comparison of vanilla retriever models vs LLM-augmented retriever performance on LoTTE datasets.

in the original documents. The average ratio of synthetic query tokens to original document tokens ($N_{T_{q^*}}/N_{T_D}$) for BEIR dataset is 100% and this ratio decreases to 58% when the subsets of Quora, HotpotAQ, MSMARCO and DBPedia are excluded. $N_{T_{q^*}}/N_{T_D}$ for LoTTE is 55%. While the number of generated tokens is comparable to that of the original tokens, our method involves only a single decoding (generation) and encoding (retrieval index construction) step throughout the entire procedure. Furthermore, our method does not require any further training, rendering it costing less than traditional query augmentation techniques that rely on augmented queries solely for retriever model training. In addition, the inference speed remains unaffected, as the retrieval index is pre-constructed using the augmented tokens.

We also compute the query match ratio, denoted as Match(q^*), which is defined as the ratio of the number of intersections between search queries and synthetic relevant queries to the total number of search queries. This metric is reported in Table 3. It is observed that most Match(q^*) values are zero, with the exceptions being the FIQA, Quora, FEVER and HopotQA subsets.

3.5 Comparative Analysis of different LLMs for Synthetic Query Generation

This section presents a comprehensive investigation into the impact of different LLMs on synthetic query generation and their subsequent effects on retrieval performance. Specifically, we compare

		Original Documents			Generated Synthetic Relevant Queries					
Dataset	Subset	N_D (in K)	N_{T_D} (in K)	N_{T_D}/N_D	N_{q^*} (in K)	$N_{T_{q^*}}$ (in K)	N_{q^*}/N_D	$N_{T_{q^*}}/N_D$	$N_{T_{q^*}}/N_{q^*}$	Match(q^*) %
BEIR	ArguAna	9	1,782	205	46	684	5	79	15	0
	FiQA	58	9,470	164	305	4,360	5	76	14	1.0
	Quora	523	8,404	16	3,123	40,947	6	78	13	6.2
	SCIDOCS	25	5,365	212	160	2,580	6	102	16	0
	SciFact	5	1,548	299	32	618	6	119	19	0
	CQADupstack	457	94,394	206	2,428	40,789	5	89	17	0
	Climate-FEVER	5,417	625,083	115	33,471	553,419	6	102	17	0
	FEVER	5,417	625,075	115	31,571	518,917	6	96	16	0.9
	HotpotQA	5,233	342,517	65	32,972	535,565	6	102	16	6.2
	MSMARCO	8,842	695,270	79	57,288	878,871	6	99	15	0
	DBPedia	4,636	331,480	72	27,023	419,920	6	91	16	0
	Touche-2020	383	85,134	223	2,491	36,333	7	95	15	0
	NQ	2,681	279,593	104	16,616	260,766	6	97	16	0
	NFCorpus	4	1,155	318	21	360	6	99	17	0
	TREC-COVID	171	36,819	215	1,027	17,196	6	100	17	0
LoTTE	Lifestyle	119	21,639	181	664	9,866	6	83	15	0
	Recreation	167	26,988	162	902	13,215	5	79	15	0
	Science	1,694	400,544	236	8,461	159,901	5	94	19	0
	Technology	662	117,940	178	7,031	105,610	11	159	15	0
	Writing	200	29,031	145	1,027	15,364	5	77	15	0

Table 3: Statistics on original document information and augmented document information for each dataset

the performance of four distinct LLMs: Llama2-7b, Llama2-70b, Llama3-8b, and Llama3-70b. The evaluation results are summarized in Table 4, which provides an overview of the R@3 and nDCG@10 performance on two BEIR datasets.

Our analysis reveals that the patterns of queries generated by different LLMs exhibit minimal variation, suggesting that the choice of LLM may not significantly influence the quality of synthetic queries. Furthermore, the corresponding recall and nDCG metrics demonstrate a similar trend, indicating that the differences between LLMs have a negligible impact on the overall retrieval performance. These findings provide valuable insights into the robustness of synthetic query generation across various LLM architectures and sizes. As a result, we opt to use smaller models (e.g. Llama3-8B) for queries and titles generation for the considerations cost-effectiveness.

3.6 Effect of Synthetic Relevant Queries and Titles

This section investigates the effect of LLM-augmented document fields, specifically synthetic query and title, on the retrieval performance of various models. We conduct a systematic analysis by manipulating field weights for Bi-encoders (Contriever and DRAGON) and isolating individual fields for the token-level late-interaction model (ColBERTv2). The experiments are performed on the LoTTE dataset, with results summarized in Table 5.

Our findings indicate that synthetic queries play a crucial role in enhancing recall for Contriever, whereas titles have a more significant impact on DRAGON's performance. For ColBERTv2, synthetic queries are found to be more influential than titles. Notably, integrating multiple document fields into a weighted sum generally improves performance across all three models, as evidenced by the comparison with recall performance in Table 2. This suggests that incorporating diverse document fields can lead to more effective retrieval outcomes.

3.7 Training-free Augmentation vs Finetuning

This section presents a comprehensive evaluation of the proposed training-free LLM-augmented retrieval method against domain-finetuned retrievers, which utilize augmented queries solely for finetuning retriever models to improve domain-specific performance. We conduct experiments on several BEIR datasets to compare the performance of these two approaches.

In our evaluation, we employ the same LLM-generated queries used in document-level embedding to create positive and negative labels for supervised training of the finetuned retrievers. Each finetuned retriever is trained exclusively on synthetic queries generated within its respective domain, ensuring a fair comparison. The training protocol involves one epoch with a learning rate of $1e-5$. The comprehensive results are presented in Table 6.

Our findings indicate that the proposed training-free LLM-augmented retrieval method is comparable to, or often surpasses, the finetuned method,

Model	BEIR	Metrics	Llama2-7b	Llama2-70b	Llama3-8b	Llama3-70b
Contriever	SciFact	R@3	58.7	60.0	60.0	62.3
		nDCG@10	56.3	58.1	58.0	60.1
Dragon	SciFact	R@3	65.2	65.4	65.8	66.7
		nDCG@10	67.8	67.5	68.0	68.6

Table 4: Comparison on synthetic relevant queries generated by different LLM models

R@3		Contriver		Dragon		ColbertV2	
LoTTE		Query Only	Title Only	Query Only	Title Only	Query Only	Title Only
Lifestyle	Search	69.7	49.0	72.5	76.1	74.1	62.2
	Forum	61.9	53.1	65.8	69.1	71.0	60.0
Recreation	Search	44.4	37.9	60.7	64.7	65.8	54.9
	Forum	53.6	49.8	58.4	62.9	64.8	52.1
Science	Search	29.0	19.0	36.5	44.1	43.3	37.0
	Forum	23.4	23.5	27.1	35.7	36.3	31.3
Technology	Search	33.0	26.7	48.7	54.4	45.3	47.2
	Forum	35.2	34.7	38.9	46.2	36.4	43.4
Writing	Search	54.7	48.1	65.8	69.3	72.7	57.8
	Forum	58.6	53.2	62.3	65.5	68.4	54.3

Table 5: Ablation study of using query only or title only on LLM-augmented retriever performance on LoTTE datasets.

while significantly reducing human effort and computational costs. This suggests that the proposed method can achieve competitive performance without the need for extensive training data or computational resources. We hypothesize that overfitting may contribute to the suboptimal performance observed in domain-finetuned models. Overfitting occurs when a model becomes too specialized to the training data and fails to generalize well to new, unseen data. In this case, the finetuned retrievers may be overfitting to the synthetic queries generated within their respective domains, leading to reduced performance on real user queries.

4 Related Work

4.1 Embedding-based Retrieval

Recent advancements in the field of information retrieval have seen the integration of neural network architectures to compute text embeddings, which have shown to outperform the traditional sparse bag-of-words models in terms of effectiveness (Dai and Callan, 2019; Luan et al., 2021). Expanding on this foundation, Liu and Croft (2002) and Bendersky and Kurland (2008) have explored paragraph-based and window-based methods to delineate passages in information retrieval, respectively. Within the neural network domain, Fan et al. (2018) illustrated that aggregating representations to assess passage-level relevance yields promising results, particularly with pre-BERT models. Furthermore, Li et al. (2023a) introduced the technique of max-pooling to evaluate passage relevance. Our methodology draws upon similar principles to these preceding studies, aiming to further refine, aggregate and enhance the information from the documents for embedding-based retrieval, through both max-pooling and average methods.

4.2 Data Augmentation and Pseudo Queries Generation

Data augmentation is a widely used technique in information retrieval training. Contrastive Learning (Izacard et al., 2021) has introduced techniques such as inverse cloze tasks, independent cropping, and random word deletion, replacement, or masking to enrich the diversity of training data. In training the DRAGON model, Lin et al. (2023) studied query augmentation using query generation models and label augmentation methods with diverse supervision.

Pre-generated pseudo queries have been shown to be effective in improving retrieval performance. Previous works have calculated the similarity between pseudo-queries and user-queries using BM25 or BERT models to determine the final rel-

BEIR	Metrics	Contriver		Dragon	
		LLM-Aug	FT	LLM-Aug	FT
ArguAna	R@3	30.3	31.2	49.8	42.2
	nDCG@10	33.2	30.1	50.3	36.7
FiQA	R3	28.7	35.4	43.8	12.3
	nDCG@10	28.7	34.6	42.5	12.5
Quora	R@3	84.9	87.2	92.1	91.6
	nDCG@10	83.1	83.6	88.3	88.1
SCIDOCS	R@3	24.3	23.1	32.1	18.9
	nDCG@10	24.6	23.2	31.8	19.5
SciFact	R@3	60.1	59.0	70.2	52.2
	nDCG@10	58.0	57.4	68.0	49.5

Table 6: The performance comparison of training-free LLM-augmented retriever vs domain-finetuned retriever

evance score of the query to document through relevance score fusion (Chen et al., 2021; Wen et al., 2023). An alternative method for generating pseudo queries involves generating pseudo query embeddings through K-means clustering algorithms (Tang et al., 2021) or some fine-tuned models (Li et al., 2023b). Large pre-trained language models have demonstrated their ability to generate high-quality text data (Anaby-Tavor et al., 2020; Kumar et al., 2020; Meng et al., 2022; Schick and Schütze, 2021; Papanikolaou and Pierleoni, 2020; Yang et al., 2020). Some previous works have leveraged the generation capabilities of language models to create synthetic training data for retriever models finetuning (Bonifacio et al., 2022; Jeronymo et al., 2023; Nogueira et al., 2019; Wang et al., 2023). In our research, we employ large language models to generate pseudo queries similarly; however, these synthetic queries are utilized not during the training phase but at the inference stage of the retrieval system, specifically pre-calculated for the construction of the retrieval index. Our approach is training-free, requiring no finetuning, and leverages the foundational knowledge of LLMs for query generation, as well as the existing capability of retrievers for calculating similarity scores. By eliminating the need for training, we can minimize costs and ensure that the method generalizes effectively across various scenarios.

5 Conclusion

This paper presents a model-agnostic and training-free framework for information retrieval, termed LLM-augmented retrieval, which significantly enhances the performance of existing retriever models. By leveraging document-level embeddings that capture contextual information derived from LLM-generated synthetic queries, titles, this approach demonstrates adaptability across various retriever model architectures. Empirical evaluations on multiple models and datasets have yielded state-of-the-art results, substantiating the efficacy of LLM-augmented retrieval in improving information retrieval quality and generalizing to new domains.

Future research directions may include refining the proposed framework by incorporating more diverse contextual information into document-level embeddings, exploring sophisticated measures for similarity scoring, and developing complex methods for integrating multiple chunks or queries into a single field embedding. These potential avenues for further investigation hold promise for continued advancements in the field of information retrieval.

6 Limitations

This study encounters several limitations, notably the increased computational resources required in generating relevant queries and titles for the original documents. In some instances, the size of the augmented texts may approach or equal that of the original documents, which could pose a significant computational burden. This limitation may hinder the applicability of this approach in environments where computational resources are constrained.

Another potential limitation concerns the risk of hallucination in large language models, which can introduce inaccuracies into the augmented corpus relative to the original documents. Hallucination remains a persistent challenge in the field of large language model research and could compromise the integrity of the retrieval process.

7 Disclaimer of AI Assistant

AI assistants (ChatGPT and Llama) are used in coding and writing of this paper.

References

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. *International Conference on Learning Representations*.

Michael Bendersky and Oren Kurland. 2008. Utilizing passage-based language models for document retrieval. In *Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30*, pages 162–174. Springer.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 169–174.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. 2023. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Xuanang Chen, Ben He, Kai Hui, Yiran Wang, Le Sun, and Yingfei Sun. 2021. Contextualized offline relevance weighting for efficient and effective neural retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1617–1621.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988.

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*.

Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. 2018. Modeling diverse relevance patterns in ad-hoc retrieval. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 375–384.

Paulo Finardi, Leonardo Avila, Rodrigo Castaldoni, Pedro Gengo, Celio Larcher, Marcos Piau, Pablo Costa, and Vinicius Caridá. 2024. The chronicles of rag: The retriever, the chunk and the generator. *arXiv preprint arXiv:2401.07883*.

Georg Gottlob, Giorgio Orsi, and Andreas Pieris. 2014. Query rewriting and optimization for ontological databases. *ACM Transactions on Database Systems (TODS)*, 39(3):1–46.

Yaakov HaCohen-Kerner, Daniel Miller, and Yair Yigal. 2020. The influence of preprocessing on text classification using a bag-of-words representation. *PloS one*, 15(5):e0232525.

Yunlong He, Jiliang Tang, Hua Ouyang, Changsung Kang, Dawei Yin, and Yi Chang. 2016. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1443–1452.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613.

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

Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. Inpars-v2: Large language models as efficient dataset generators for information retrieval. *arXiv preprint arXiv:2301.01820*.

William P Jones and George W Furnas. 1987. Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American society for information science*, 38(6):420–442.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2023a. Parade: Passage representation aggregation fordocument reranking. *ACM Transactions on Information Systems*, 42(2):1–26.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023b. Multiview identifiers enhanced generative retrieval. *arXiv preprint arXiv:2305.16675*.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Xiaoyong Liu and W Bruce Croft. 2002. Passage retrieval based on language models. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 375–382.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660.

Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to doctttttquery. *Online preprint*, 6(2).

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

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

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2021. Colbertv2: Effective and efficient retrieval via lightweight late interaction. *arXiv preprint arXiv:2112.01488*.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. *arXiv preprint arXiv:2104.07540*.

Jagendra Singh and Aditi Sharan. 2017. A new fuzzy logic-based query expansion model for efficient information retrieval using relevance feedback approach. *Neural Computing and Applications*, 28:2557–2580.

Hongyin Tang, Xingwu Sun, Beihong Jin, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. Improving document representations by generating pseudo query embeddings for dense retrieval. *arXiv preprint arXiv:2105.03599*.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Xueru Wen, Xiaoyang Chen, Xuanang Chen, Ben He, and Le Sun. 2023. Offline pseudo relevance feedback for efficient and effective single-pass dense retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2209–2214.

Chenyan Xiong and Jamie Callan. 2015. Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 111–120.

Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey.

2020. Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546*.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1:43–52.

Appendix

I will give you an article below. What are some search queries or questions that are relevant for this article or this article can answer? Separate each query in a new line. This is the article: {document} Only provide the user queries without any additional text. Format every query as 'query:' followed by the question. Don't write empty queries.

Table 7: Prompt for generating relevant queries for documents

I will give you an article below. Create a title for the below article. This is the article: {document} Only provide the title without any additional text. Format the reply starting with 'title:' followed by the question. Don't write empty title.

Table 8: Prompt for generating titles for documents.