Large Language Models as Foundations for Next-Gen Dense Retrieval: A Comprehensive Empirical Assessment

Kun Luo[1,2,3†] **Minghao Qin**[2†] **Zheng Liu**[2*] **Shitao Xiao**[2] **Jun Zhao**[1,3] **Kang Liu**[1,2,3*]
[1]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]Beijing Academy of Artificial Intelligence, Beijing, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
{luokun695, zhengliu1026}@gmail.com kliu@nlpr.ia.ac.cn

Abstract

Pre-trained language models like BERT and T5 serve as crucial backbone encoders for dense retrieval. However, these models often exhibit limited generalization capabilities and face challenges in improving in-domain accuracy. Recent research has explored using large language models (LLMs) as retrievers, achieving state-of-the-art performance across various tasks. Despite these advancements, the specific benefits of LLMs over traditional retrievers and the impact of different LLM configurations—such as parameter sizes, pre-training duration, and alignment processes—on retrieval tasks remain unclear.

In this work, we conduct a comprehensive empirical study on six key dimensions of dense retrieval capabilities, including in-domain accuracy, data efficiency, zero-shot generalization, lengthy retrieval, instruction-based retrieval, and multi-task learning. We evaluate over 15 different backbone LLMs and non-LLMs. Our findings reveal that larger models and extensive pre-training consistently enhance in-domain accuracy and data efficiency. Additionally, larger models demonstrate significant potential in zero-shot generalization, lengthy retrieval, instruction-based retrieval, and multi-task learning. These results underscore the advantages of LLMs as versatile and effective backbone encoders in dense retrieval, providing valuable insights for future research and development in this field.

1 Introduction

Dense retrieval, a novel paradigm in Information Retrieval (IR), has emerged with the advancement of deep neural networks. Unlike traditional IR methods, dense retrieval encodes both queries and documents as embeddings within a shared latent space, capturing their semantic relationships through embedding similarities. Dense retrieval models have become the predominant choice in recent neural retrieval approaches and are widely applied in various downstream tasks such as web search, question answering, and sentence similarity (Karpukhin et al., 2020; Xiong et al., 2020; Muennighoff et al., 2022).

In the past few years, dense retrieval models intensively adopted pre-trained language models, such as BERT (Devlin et al., 2018) and T5 (Raffel et al., 2020), as their backbone encoders. These models excel in identifying semantic similarities between queries and documents. However, they still face significant challenges in becoming versatile enough to handle a wide range of retrieval tasks (Muennighoff et al., 2022). Their in-domain retrieval accuracy is often constrained by the capacity of their backbone encoders, such as the number of parameters (Ni et al., 2021). Additionally, dense retrieval models typically struggle to generalize to unseen data, necessitating fine-tuning with a large amount of labeled data to perform well in the target domain. Finally, achieving versatility in dense retrieval models requires training on multiple retrieval tasks simultaneously, which demands sufficient capacity from the backbone encoder (Zhang et al., 2023; Xiao et al., 2023).

Recently Large Language Models (LLMs) have been prompted or fine-tuned as dense retrieval models and achieved improved performance across a wide range of retrieval tasks, thanks to their superior capability for semantic understanding and rich world knowledge (Li et al., 2023; Wang et al., 2023; Zhuang et al., 2024; Muennighoff et al., 2024). These models vary in parameters from 2 billion to 56 billion, with pre-training sufficiency ranging from hundreds of billions to tens of trillions of tokens, and include both base models and human preference aligned chat models. Despite the common understanding that larger models generally yield better performance (Kaplan et al., 2020; Biderman et al., 2023), the specific benefits of vary-

†. Equal contribution
*. Corresponding author

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 1354–1365
November 12-16, 2024

ing parameter numbers, pre-training sufficiency, and alignment processes of backbone LLMs for different retrieval tasks still remain unclear.

In this study, we focus on the following two research questions: 1) For different retrieval tasks, what specific benefits can LLMs offer compared to non-LLMs as the backbone encoders? 2) For LLMs with varying configurations (i.e., different parameter numbers, pre-training sufficiency and alignment processes), what contributes more to different retrieval tasks as the backbone encoder. We conduct comprehensive empirical investigation across a wide range of retrieval tasks, assessing various critical retrieval capabilities: in-domain accuracy, data efficiency, zero-shot generalization, lengthy retrieval generalization, instruction-based retrieval, and multi-task learning. Our study explore over 15 different backbone LLMs and non-LLMs, with parameter numbers ranging from 0.1 billion to 32 billion and varying pre-training sufficiency, including both base LLMs and chat LLMs.

Previous dense retrieval models have demonstrated inferior **in-domain accuracy** due to the limited capacity of their backbone encoders (Ni et al., 2021). We employ MS MARCO (Nguyen et al., 2016), one of the largest web search datasets, to train and evaluate the in-domain accuracy of dense retrieval models with different backbone encoders. Our results indicate that both increasing the model size and enhancing pre-training sufficiency can consistently improve the upper limit of in-domain accuracy. Notably, we discover that both base LLMs and human-preference-aligned chat LLMs show comparable potential as backbone encoders for dense retrieval tasks. By training with different proportions of MS MARCO, we explore **data efficiency** and find that scaling up model size facilitates convergence, allowing LLMs to converge swiftly even with limited annotated data, without the need for intricate multi-stage training processes.

We examine generalization ability from three perspectives: zero-shot generalization, lengthy retrieval generalization, and instruction-based retrieval generalization. First, we evaluate **zero-shot generalization** using BEIR benchmark (Thakur et al., 2021). Our findings indicate that model size is the most crucial factor for zero-shot retrieval generalization. Moreover, traditional dense retrieval models are limited by the maximum input length used during pre-training and retrieval training. We investigate whether LLM-based retrievers, pre-trained with longer context windows, can effectively generalize to **lengthy retrieval** tasks even when trained with shorter passage lengths. Finally, dense retrieval models often lack flexibility in handling varying retrieval intents (Su et al., 2022). We explore the capability of different models to **incorporate instructions** during retrieval, discovering that training with instruction benefits LLMs but not non-LLMs, and that human-preference alignment does not significantly improve performance compared to base LLMs.

We further explore the **multi-task learning** capabilities of models with different backbone encoders, essential for developing versatile retrievers (Zhang et al., 2023; Xiao et al., 2023). We adopt five distinct retrieval tasks, where interference exists due to varying retrieval intents. Our findings reveal that although all models experience performance decreases with multi-task training compared to training on each single-task, increasing model size consistently mitigates this gap.

To summarize, we make the following contributions: 1) We conduct a thorough experimental study using more than 15 backbone encoders with different configurations for dense retrieval across six distinct retrieval tasks. 2) We demonstrate that LLM-based retrievers consistently enhance performance across all retrieval tasks compared to non-LLM-based retrievers. 3) We investigate how different configurations of backbone LLMs impact each retrieval task, focusing on distinct retrieval capabilities.

2 Related Work

The related works are reviewed from two aspects: dense retrieval, LLM-based retriever.

First of all, in the realm of neural retrievers, dense retrieval models have consistently demonstrated superior performance over traditional sparse models like BM25 across a wide array of retrieval tasks (Karpukhin et al., 2020; Ni et al., 2021; Muennighoff et al., 2022). A critical factor contributing to the success of dense retrieval models is the utilization of powerful pre-trained language models as their initialization.

Over the past few years, pre-trained language models such as BERT (Devlin et al., 2018) and T5 (Raffel et al., 2020) have been intensively used as backbone encoders for dense retrieval. For instance, GTR (Ni et al., 2021) highlights the in-

domain accuracy and generalization capabilities of T5-based dense retrieval models, with model parameters reaching up to 4.8 billion. Fang et al. (2024) explores scaling laws for dense retrieval models but restricts their study to BERT backbones with up to 110 million parameters and only explores the in-domain situation. Currently, state-of-the-art dense retrievers employ models with more than 7 billion parameters or more as backbones. Neelakantan et al. (2022) discuss large-scale unsupervised text embedding pre-training, observing consistent performance improvements when scaling up GPT-based dense retrieval model sizes from 300 million to 175 billion parameters. Additionally, recent studies such as Wang et al. (2023) have shown that fine-tuning directly with labeled data can achieve strong performance. Our study focuses on fine-tuning directly using labeled data while comparing various backbone encoders.

Large Language Models (LLMs) have recently demonstrated significant potential as backbone encoders for dense retrieval, attributed to their vast number of parameters and extensive pre-training. Repllama (Ma et al., 2023) fine-tuned Llama-2-7B and Llama-2-13B to function both as dense retrievers and pointwise rerankers. LLaRA (Li et al., 2023) introduced two pretraining tasks specifically designed to better adapt the backbone Llama-2-7B model for dense retrieval, resulting in notable improvements in both supervised and zero-shot scenarios. E5-mistral and Gecko (Wang et al., 2023; Lee et al., 2024) enhanced the training of LLM-based dense retrievers using synthetic data, employing models with 1.5 billion and 7 billion parameters to achieve notable results across various retrieval tasks. GRIT (Muennighoff et al., 2024) successfully unified text embedding and generation within a single LLM, maintaining performance levels comparable to those of specialized embedding-only and generative-only models, using a model with 56 billion parameters (14 billion activation parameters). LLM2Vec (BehnamGhader et al., 2024) presented an unsupervised method for transforming decoder-only LLMs into dense retrievers, demonstrating significant promise for adapting LLM backbone encoders for dense retrieval in an unsupervised manner. PromptReps (Zhuang et al., 2024) employed human preference-aligned chat LLMs to produce high-quality dense representations unsupervised.

These models vary in parameters from 1.5 billion to 56 billion, with pre-training covering hundreds of billions to tens of trillions of tokens, and include both base LLMs and human preference-aligned chat LLMs. Despite the exciting advancements in retrieval tasks achieved by leveraging various LLMs with distinct configurations and diverse training strategies, the specific benefits of variations in parameter count, pre-training extent, and alignment processes of backbone LLMs for retrieval tasks remain still uncertain.

3 Preliminary

Dense retrieval leverages an encoder to project both the query q and the candidate passage p into a shared dense embedding space, resulting in embeddings h_q and h_p. A scoring function, such as the inner product or cosine similarity, is then applied to these dense vectors to model relevance:

$$\mathrm{s}(\mathrm{q}, \mathrm{p}) = \langle \mathrm{h}_q, \mathrm{h}_p \rangle \tag{1}$$

This allows for the retrieval of relevant documents by performing approximate nearest neighbor (ANN) search within the embedding space.

In our study, we compare more than 15 backbone encoders, varying in model architecture (encoder-only and decoder-only), model size (0.1B to 32B), and pre-training sufficiency (up to 15T tokens). Consistent with prior research, we utilize the [CLS] token to obtain text representations for the BERT model and employ mean-pooling for the T5 model. For instance, BERT tokenizes the input text into a sequence T: [CLS], t_1, ..., t_N, [EOS]. This tokenized sequence is subsequently encoded by BERT, generating output embeddings that are combined to form the text embedding, with the [CLS] token performing this integration:

$$\mathrm{h}_t = \mathrm{BERT}(\mathrm{T})[\mathrm{CLS}] \tag{2}$$

When using large language model (LLM) as the backbone encoder, text embeddings need to be created differently. Most LLMs use a decoder-only architecture and causal attention mechanism, meaning that only the last token in the input sequence can access the global context. As a result, the text embedding is taken from the output embedding of the special token [EOS]:

$$\mathrm{h}_t = \mathrm{LLM}(\mathrm{T})[\mathrm{EOS}] \tag{3}$$

Given the query-passage pair $(\mathrm{q}_i, \mathrm{p}_i^+)$, we adopt the standard InfoNCE (Izacard et al., 2021) loss L

over the in-batch negatives and hard negatives for training:

$$\mathrm{L} = -\lg \frac{\exp(\mathrm{s}(\mathrm{q}_i, \mathrm{p}_i^+))}{\exp(\mathrm{s}(\mathrm{q}_i, \mathrm{p}_i^+)) + \sum_j \exp(\mathrm{s}(\mathrm{q}_j, \mathrm{p}_j^-))} \tag{4}$$

where p_j^- is the set of negative passages and $\mathrm{s}(\mathrm{q}, \mathrm{p})$ is the scoring function of query and passage. In this paper, we adopt the temperature-based cosine similarity function as follows:

$$\mathrm{s}(\mathrm{q}, \mathrm{p}) = \frac{1}{\tau} \cos(\mathrm{h}_q, \mathrm{h}_p) \tag{5}$$

τ is a temperature hyper-parameter, which is fixed to 0.02 in all experiments.

4 Empirical Study

In this section, we aim to address two key research questions: 1) For different retrieval tasks, what specific benefits can LLMs offer compared to non-LLMs as the backbone encoders? 2) For LLMs with varying configurations (i.e., different parameter numbers, pre-training sufficiency, and alignment processes), what contributes more to different retrieval tasks as the backbone encoder. To answer these questions, we conduct a comprehensive empirical study across six critical dimensions of dense retrieval, each encompassing several specific retrieval tasks. These dimensions are investigated using various pre-trained language models as backbone encoders, focusing on: in-domain accuracy (Section 4.1), data efficiency (Section 4.2), zero-shot generalization (Section 4.3), lengthy retrieval generalization (Section 4.4), instruction-based retrieval (Section 4.5), and multi-task learning (Section 4.6).

4.1 In-domain Accuracy

Setting We utilize MS MARCO (Nguyen et al., 2016) to train and evaluate the in-domain accuracy of dense retrieval models with varying backbones encoders. Specifically, we employ BERT (Devlin et al., 2018) with 110M and 330M parameters (BERT-base and BERT-large), T5 (Raffel et al., 2020) encoders with parameter numbers ranging from 110M to 4.8B, and a diverse set of LLMs including the Llama, Phi, Gemma, and Qwen1.5 series (Touvron et al., 2023; Gunasekar et al., 2023; Bai et al., 2023; Team et al., 2024). It is important to note that different LLMs have varying configurations. For instance, the phi-1.5 model is a lightweight LLM with 1.3B parameters and is pre-trained on a relatively small amount of tokens (150B), indicating less pre-training sufficiency. In contrast, the Llama-3-8B model is extensively pre-trained on over 15T tokens, significantly more than the 2T tokens used for Llama-2-7B. The Qwen1.5 series offers a variety of models in different sizes, all pre-trained on the same corpus, enabling direct comparisons of the effects of scaling up model size.

All models are trained with a batch size of 128 and incorporate 7 hard negative samples to ensure fair comparisons of in-domain retrieval accuracy. All training operations take place on 8xA800 (80GB) GPUs. We use the Adam optimizer with an initial learning rate of 3e-4 and linear decay. For training LLM retrievers, we employ LoRA (Hu et al., 2021), which has demonstrated similar efficacy to full-parameter fine-tuning for retrieval tasks (Ma et al., 2023). The in-domain accuracy of each model is evaluated using the MS MARCO development set, comprising 6,980 queries. We use NDCG@10, MRR@10, Recall@10, and Recall@1000 as evaluation metrics, providing a comprehensive analysis of in-domain performance.

Results and Analysis As presented in Figure 1, the results indicate that model performance generally improves with an increase in parameter numbers. This trend is particularly noticeable within models from the same series. For instance, the Qwen1.5 series demonstrates this progression: Qwen1.5-0.5B model scores 36.7, while the Qwen1.5-32B model achieves 42.6, representing an improvement of 5.9 points. This trend suggests that increasing model size is a feasible way to yield better in-domain accuracy. Detailed results are presented in Table 5.

Additionally, the results demonstrate that LLM-based retrievers significantly outperform non-LLM retrievers. The performance of Gemma-2B has already surpassed all BERT and T5-based models despite having fewer parameters than the T5-xxl model. This suggests that LLMs' extensive pre-training and advanced language understanding capabilities offer significant advantages as backbone encoders for dense retrieval.

An interesting observation is that smaller models can sometimes marginally outperform larger ones. The Qwen1.5-0.5B model, with fewer parameters, surpasses the Phi-1.5-1.3B model and competes closely with the Phi-2-2.7B model. This performance discrepancy may be attributed to differences in pre-training sufficiency. The Qwen1.5

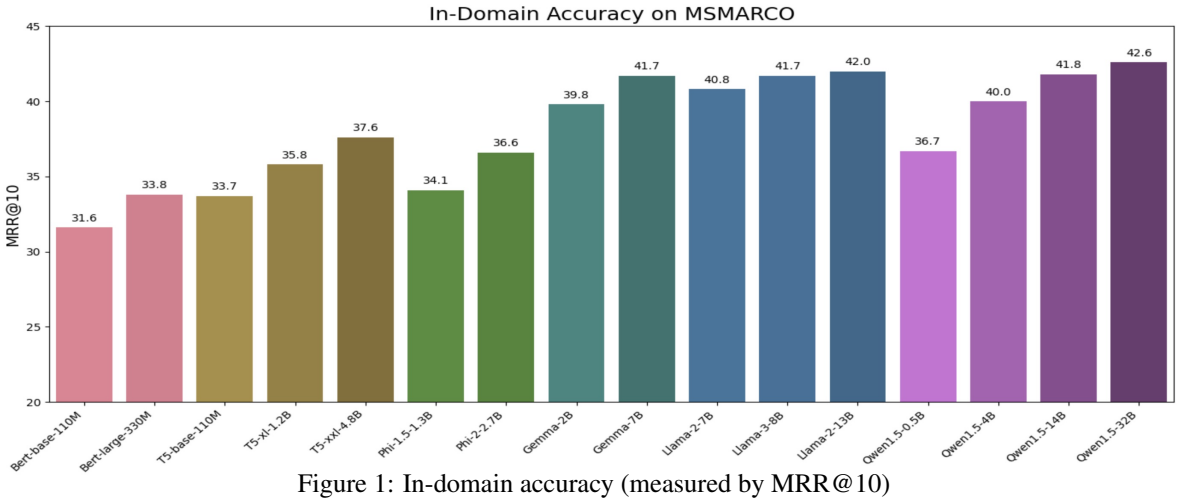

Figure 1: In-domain accuracy (measured by MRR@10)

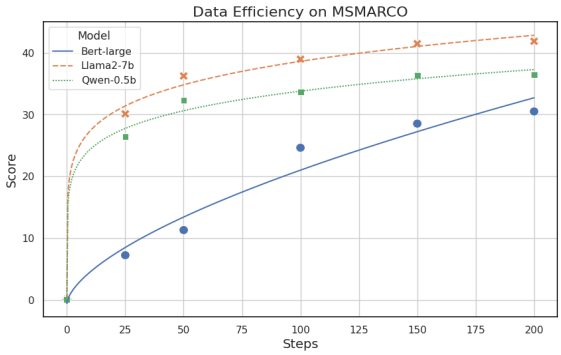

Figure 2: Data efficiency

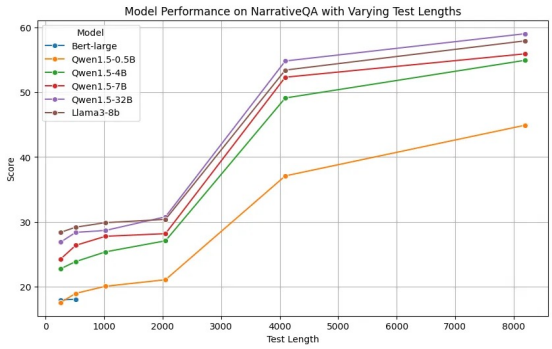

Figure 3: Lengthy retrieval

models benefit from more extensive and diverse pre-training data, totaling over 3 trillion tokens, whereas the Phi models are pre-trained on a smaller amount of high-quality data, with 150 billion tokens for the Phi-1.5 and 1.4 trillion tokens for the Phi-2. This extensive pre-training enables the Qwen1.5-0.5B model to perform better when fine-tuned for retrieval tasks. A similar conclusion can be drawn from the comparison between the Llama-3-8B and Llama-2-7B models, as well as between LLMs and non-LLMs. Extensive and varied pre-training of backbone encoders can significantly enhance in-domain retrieval accuracy, even compensating for a smaller parameter count.

4.2 Data Efficiency

Setting We use checkpoints from models trained on MS MARCO for different numbers of steps to evaluate their performance on the development set, in order to better understand the impact of parameter number and pre-training sufficiency on data efficiency and convergence speed.

We compare BERT-large, Qwen1.5-0.5B, and Llama-2-7B to explore the impact of data efficiency with model parameter number and pre-training sufficiency. Notably, BERT-large and Qwen1.5-0.5B have similar non-embedding parameter number, while Qwen1.5-0.5B is based on decoder architecture and has undergone more extensive pre-training.

Results and Analysis As presented in Figure 2, our findings indicate that larger model sizes lead to higher data efficiency and faster convergence. Specifically, after 100 training steps on MS MARCO, Llama-2-7B outperforms Qwen1.5-0.5B by 5.4 points and BERT-large by 14.4 points. This suggests that with an increase in parameter number, better performance can be achieved with less labeled data. Furthermore, as shown in Table 1, when comparing the relative score difference between 100 steps and the full training of 3700 steps, Llama-2-7B shows a score difference of 8.8 points, which is smaller than the 9.7 points for Qwen1.5-0.5B and 15.3 points for BERT-large. This indicates that larger models are able to converge faster.

The experiment results also demonstrate that LLMs have better data efficiency compared to non-LLMs, even with similar parameter sizes. For example, after 100 training steps on MS MARCO, Qwen1.5-0.5B outperforms BERT-large by 9 points. Despite having a similar number of parameters, Qwen1.5-0.5B has undergone more

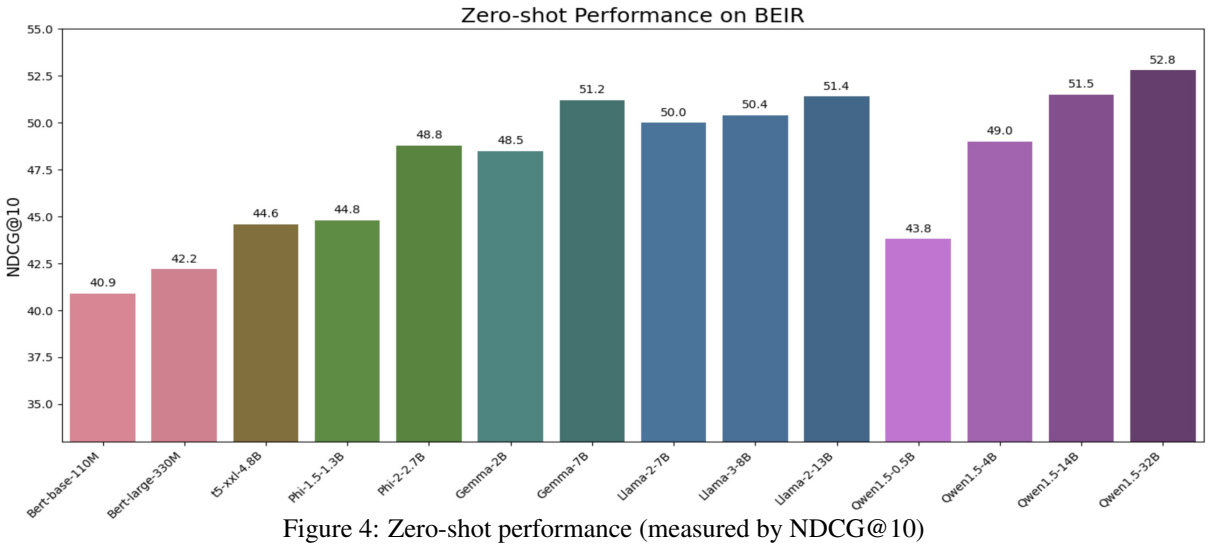

Figure 4: Zero-shot performance (measured by NDCG@10)

Model	Parameter Number	NDCG@10	MRR@10	Recall@10
100 Steps				
Bert-large	0.3 B	24.6($\delta = 15.3$)	20.0	40.5
Qwen1.5-0.5B	0.5 B	33.6($\delta = 9.7$)	27.9	53.2
Llama-2-7B	7 B	39.0($\delta = 8.8$)	32.4	61.0
Full 3700 Steps				
Bert-large	0.3 B	39.9	33.8	60.3
Qwen1.5-0.5B	0.5 B	43.3	36.7	65.5
Llama-2-7B	7 B	47.8	40.8	70.9

Table 1: Model convergence speed.

Model	Parameter Number	MSMARCO-ID	MSMARCO-OOD
Bert-large	0.3 B	40.0	39.3
Qwen1.5-0.5B	0.5 B	43.5	43.6
Qwen1.5-4B	4 B	47.0	47.0
Qwen1.5-14B	14 B	48.9	48.9
Llama-3-8B	8 B	49.6	49.6

Table 2: Unseen instruction comparison. "ID" means instructions are seen during training, "OOD" means the instructions are unseen during training.

extensive pre-training (over 3 trillion tokens compared to BERT's 3.3 billion tokens) and employs a decoder architecture, which enhances its language understanding ability and enables faster convergence in the retrieval task where text discriminative ability is crucial.

4.3 Zero-Shot Generalization

Setting Dense retrieval models typically struggle with zero-shot retrieval on unseen data (Ni et al., 2021). We investigate the specific benefits that LLM-based retrievers can bring to zero-shot generalization, focusing on varying model sizes and pre-training sufficiency.

We evaluate all models on 13 zero-shot retrieval tasks in the BEIR (Thakur et al., 2021) evaluation suite, which encompasses a diverse range of retrieval tasks and domains, including medical retrieval, financial retrieval, and duplication detection. All models are directly transferred for zero-shot evaluation on BEIR after being trained on MS MARCO. During the evaluations, we set the maximum length of the query to 64 tokens and the maximum length of the passage to 256 tokens.

Results and Analysis The results are shown in Figure 4, measured by average performance of NDCG@10 across 13 retrieval tasks. LLM retrievers significantly outperform non-LLM retrievers in zero-shot retrieval tasks, indicating that the extensive knowledge and robust generalization capabilities of LLMs are highly advantageous for zero-shot retrieval. Notably, this improvement is not merely a result of increased model size: even the Qwen1.5-0.5B model, which has a similar non-embedding parameter count, demonstrates much better generalization (+1.6%) than the BERT-large model. This highlights the potential of LLMs to serve as robust encoders for various retrieval domains.

For different configurations of LLMs, model size is the primary factor influencing their generalization capability. Unlike in-domain accuracy, where both model size and pre-training sufficiency are important, generalization performance is almost directly correlated with the number of parameters. For example, the Qwen-0.5B model, despite benefiting from more extensive pre-training, performs worse than the Phi-1.5-1.3B and Phi-2-2.7B models with larger parameter sizes but less pre-training sufficiency. This suggests that larger models, with better capacity, can prevent overfitting to domain-specific retrieval data, resulting in better generalization to unseen data.

4.4 Lengthy Retrieval Generalization

Setting Traditional dense retrieval models are constrained by the maximum input length used during

Model	Hotpot	NQ	MSM	FiQA	NFCorpus	SciFact	Average
BERT-large	46.8(-4.6)	47.3(+0.9)	40.0(+0.1)	24.3(-2.0)	24.7(-2.0)	55.5(+0.9)	39.8(-1.0)
Qwen1.5-0.5B	59.3(+2.7)	50.5(+7.1)	43.5(+0.2)	33.5(-0.4)	31.8(+1.5)	66.2(-0.6)	47.4(+1.7)
Qwen1.5-4B	63.6(-0.1)	57.7(+7.4)	47.0(+0.2)	39.8(+0.4)	34.8(-0.6)	72.1(+1.3)	52.5(+1.4)
Qwen1.5-14B	69.5(+3.2)	63.0(+3.7)	48.9(+0.2)	45.6(+0.6)	37.0(+0.6)	75.9(+1.7)	56.7(+1.8)
Llama-3-8B	70.9(+4.9)	63.1(+6.7)	49.6(+0.9)	44.8(+3.1)	37.8(+2.6)	75.4(+1.4)	56.8(+3.2)
Qwen1.5-0.5B-Chat	57.5	49.5	43.6	32.8	31.7	65.0	46.7
Qwen1.5-4B-Chat	64.0	58.1	47.2	40.2	36.1	71.3	52.8
Qwen1.5-14B-Chat	69.4	63.5	49.0	44.4	37.1	76.0	56.6
Llama-3-8B-Chat	70.6	63.0	49.6	44.8	38.2	75.5	56.9

Table 3: Instruction-based retrieval performance measured by NDCG@10. The average performance discrepancy is compared to training without instruction.

Model	Hotpot	STS	MSM	Tool	QReCC	Average
BERT-large	62.1(-2.4)	80.2(+2.7)	38.8(-1.1)	76.6(-5.2)	47.3(-4.1)	61.0(-2.0)
Qwen1.5-0.5B	72.1(-1.5)	80.1(+1.0)	43.7(+0.2)	84.8(-4.8)	50.7(-3.9)	66.3(-1.8)
Qwen1.5-4B	79.8(-0.6)	82.0(+2.2)	46.8(+0.0)	86.1(-4.2)	54.9(-4.4)	69.9(-1.4)
Llama-3-8B	85.7(+0.3)	82.8(+1.3)	48.9(+0.2)	89.9(-2.7)	59.5(-3.3)	73.4(-0.8)

Table 4: Multi-task learning performance measured by NDCG@10. The performance discrepancy is compared to training on each single task.

pre-training and retrieval training, while extending this length significantly increases computational costs (Chen et al., 2024). Given that LLMs are pre-trained with longer context windows, we investigate if they can be trained with shorter passage lengths while effectively generalizing to longer lengths during retrieval. We use MS MARCO for training and set the maximum query length to 64 tokens and the maximum passage length to 256 tokens. All other hyperparameters are aligned with those used in Section 4.1.

For evaluation, we utilize NarrativeQA (Kočiskỳ et al., 2018), which requires long context information to accurately retrieve target queries. The evaluation was conducted with maximum lengths ranging from 256 to 8192 tokens for passages, with the goal of thoroughly assessing each model's length generalization capabilities in the retrieval task.

Results and Analysis The results are illustrated in Figure 3. The long context window of LLMs improves length generalization compared to BERT. When evaluated with a context length of 256 tokens on the NarrativeQA Retrieval task, BERT-large outperforms Qwen1.5-0.5B by 0.4 points. However, with a length of 512 tokens, Qwen1.5-0.5B exceeds the performance of BERT-large by 0.9 points. This interesting finding demonstrates that LLM retrievers consistently generalize better with increasing input lengths, while non-LLM retrievers like BERT struggle with longer inputs and are constrained by a 512-token limit unless explicitly extended. Detailed results are presentend in Table 7

Furthermore, increasing the parameter number of LLM retrievers consistently enhances performance with longer inputs. This indicates that scaling up LLMs is an effective strategy for improving lengthy retrieval generalization, obviating the need for specific training on longer retrieval inputs.

4.5 Instruction-Based Retrieval

Setting Dense retrieval models often lack flexibility in adapting to varying retrieval intents of users, which is both common and critical in real-world retrieval scenarios (Su et al., 2022). We incorporate instructions into the training of dense retrieval models, aiming to evaluate the instruction comprehension capabilities of models with different backbone encoders. Specifically, we prepare five retrieval instructions and prepend them to queries during training on MS MARCO. We conduct evaluation on six retrieval tasks, including both in-domain and out-of-domain scenarios, to determine whether incorporating instructions can enhance the understanding of retrieval intent thus improving general performance of different models. The instructions are presented in Figure 5.

Results and Analysis As shown in Table 3, training with instructions significantly improves the performance of LLM retrievers, whereas for BERT retrievers results in decreased performance. This suggests that LLMs have superior semantic understanding, enabling them to adjust retrieval objectives based on instructions.

We evaluate models on MS MARCO (Nguyen et al., 2016) development set using instructions not seen during training. The result is presented in Table 2. These instructions are complex modifications of the training instructions (Figure 5), designed to test the models' robustness. The results show that LLM retrievers exhibit strong robustness

to these new instructions, while BERT experience performance degradation due to interference from the unseen instructions. This implies that LLMs can better utilize their capabilities in real-world retrieval scenarios as backbone encoder for dense retrieval, offering better customizability and adaptability to meet diverse user retrieval needs.

Furthermore, we adopt chat LLMs as backbone encoders to investigate if these aligned models could better utilize retrieval instructions, the result is shown in Table 3. Contrary to expectations, chat LLMs do not show further improvements when trained and tested under the same setting as base models. Thus, given the superior scalability of base LLMs across various downstream tasks, the base LLMs remain more suitable as backbone encoders for dense retrieval models.

4.6 Multi-Task Learning

Setting Training a versatile dense retrieval model is challenging due to the specific semantic information required by various retrieval tasks, often causing mutual interference (Zhang et al., 2023; Xiao et al., 2023; Neelakantan et al., 2022). We explore the multi-task learning capacity of different backbone encoders, which is essential for developing robust retrievers.

Our study encompasses four distinct retrieval tasks alongside a text similarity task: 1) ToolLLM (Qin et al., 2023): This task evaluates the ability of retrievers to identify necessary tools based on provided instructions and tool descriptions. Performance is measured using NDCG@5 on the test set. 2) QReCC (Anantha et al., 2020): This task involves retrieving relevant knowledge based on the concatenation of conversation context and the most recent query. Performance is assessed using NDCG@3, in line with previous studies (Mao et al., 2023). 3) NLI (Bowman et al., 2015): We utilize the NLI training set to establish text similarity capabilities and evaluate models on STS tasks from the MTEB (Muennighoff et al., 2022). 4) HotpotQA (Yang et al., 2018): This task tests retrieval performance in a multi-hop question-answering scenario. 5) MS MARCO (Nguyen et al., 2016): This task assesses the web search capabilities of different models.

Results and Analysis As shown in Table 4, the results demonstrate a clear trend: as model size increases, the average performance across the five distinct retrieval tasks improves. This indicates that larger models exhibit enhanced universality and capacity, suggesting their greater potential to serve as versatile embedding models in multi-task scenarios.

In addition to comparing the absolute performance of each model across multiple tasks, we conducted experiments contrasting the performance of models trained on each individual task versus joint multi-task training. Table 4 presents the relative performance discrepancy. We observed that multi-task training results in a relative performance decrease compared to single-task training across all tasks. This aligns with the hypothesis proposed by (Neelakantan et al., 2022), suggesting that certain retrieval tasks might have inherently conflicting definitions, such as search and sentence similarity tasks. Notably, the performance decrease diminishes as model size increases, indicating that larger models might be capable of learning the intrinsic relationships and distinctions between tasks during multi-task training. This capability potentially allows these models to narrow the performance gap between multi-task and single-task training, and in some cases even achieve improvements over single-task training. This suggests that LLMs with more parameter numbers have the potential to serve as versatile general-purpose retrievers across multiple retrieval tasks.

5 Conclusions

In this paper, we conduct a comprehensive empirical investigation into the benefits and configurations of LLMs as backbone encoders for dense retrieval tasks. Our focus is on comparing LLMs with non-LLMs and analyzing the impact of various LLM configurations, such as parameter count, pre-training sufficiency, and alignment processes. Our study highlights the significant advantages of utilizing LLMs as backbone encoders for dense retrieval tasks. We find that increasing the parameter count and ensuring sufficient pre-training of backbone encoders enhance in-domain accuracy. Additionally, adopting larger models consistently yields performance gains in zero-shot retrieval generalization, lengthy retrieval generalization, and multi-task learning. These insights provide a foundation for future research aimed at optimizing dense retrieval models by balancing model size and pre-training sufficiency of backbone LLMs to achieve superior performance across diverse retrieval scenarios.

6 Limitations

While our study provides valuable insights into the benefits and configurations of LLMs as backbone encoders for dense retrieval tasks, several limitations should be considered: Firstly, some experiments lack comparisons with all other backbone models in the same series, such as in data efficiency and multitask performance. Secondly, there are still some capability dimensions of retrieval models that haven't been examined, such as multilingual retrieval and robustness against noisy data. Additionally, certain characteristics of LLMs, such as whether they use unidirectional or bidirectional attention mechanisms, and the overlap between pretraining data and downstream retrieval task data, have not been explored. Addressing these aspects in future studies could provide a more complete, general conclusion.

7 Ethical consideration

Our research explores the use of various Large Language Models (LLMs) as backbone encoders for dense retrieval tasks. Despite undergoing additional fine-tuning in various experiments, these models retain ethical and social risks inherent in their pretraining data. Notably, open-source LLMs may incorporate private or contentious data during the training phase, thereby raising additional ethical concerns.

8 Ackonwledgements

We would like to thank all the reviewers for their helpful feedback, and EMNLP 2024 and ACL Rolling Review organizers for their efforts. This work was supported by Beijing Natural Science Foundation (L243006) and CCF-BaiChuan-Ebtech Foundation Model Fund.

References

Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. 2020. Open-domain question answering goes conversational via question rewriting. *arXiv preprint arXiv:2010.04898*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling laws for dense retrieval. *arXiv preprint arXiv:2403.18684*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

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*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, et al. 2024. Gecko: Versatile text embeddings distilled from large language models. *arXiv preprint arXiv:2403.20327*.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *arXiv preprint arXiv:2312.15503*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *arXiv preprint arXiv:2310.08319*.

Kelong Mao, Hongjin Qian, Fengran Mo, Zhicheng Dou, Bang Liu, Xiaohua Cheng, and Zhao Cao. 2023. Learning denoised and interpretable session representation for conversational search. In *Proceedings of the ACM Web Conference 2023*, pages 3193–3202.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. 2022. Text and code embeddings by contrastive pretraining. *arXiv preprint arXiv:2201.10005*.

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.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings. *arXiv preprint arXiv:2212.09741*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. 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*.

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

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *arXiv preprint arXiv:2310.07554*.

Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. 2024. Promptreps: Prompting large language models to generate dense and sparse representations for zero-shot document retrieval. *arXiv preprint arXiv:2404.18424*.

Model	Dimension	NDCG@10	MRR@10	R@10	R@1000
BERT-base	768	37.5	31.6	57.4	95.2
BERT-large	1024	39.9	33.8	60.3	96.0
T5-base	768	40.1	33.7	61.5	97.3
T5-xl	2048	42.3	35.8	64.0	98.3
T5-xxl	4096	44.2	37.6	66.2	98.6
Phi-1.5-1.3B	2048	40.6	34.1	62.2	98.0
Phi-2-2.7B	2560	43.3	36.6	65.8	98.6
Gemma-2B	2048	46.8	39.8	70.1	99.2
Gemma-7B	3072	48.7	41.7	72.1	99.4
Llama-2-7B	4096	47.8	40.8	70.9	99.4
Llama-3-8B	4096	49.0	42.1	71.9	99.5
Llama-2-13B	5120	48.7	42.0	71.4	99.5
Qwen1.5-0.5B	1024	43.3	36.7	65.5	98.2
Qwen1.5-4B	2048	46.8	40.0	69.7	99.2
Qwen1.5-14B	5120	48.3	41.3	71.5	99.4
Qwen1.5-32B	5120	49.5	42.6	72.7	99.5
Qwen1.5-0.5B-Chat	1024	43.3	36.8	65.1	98.1
Qwen1.5-4B-Chat	2048	47.0	40.1	70.0	99.2
Qwen1.5-14B-Chat	5120	48.6	41.5	71.8	99.4
Llama-3-8B-Chat	4096	48.7	41.8	71.6	99.4

Table 5: Detailed result of in-domain accuracy on MS MARCO.

Model	ArguAna	ClimateFEVER	DBPedia	FEVER	FiQA2018	HotpotQA	NFCorpus	NQ	Quora	SCIDOCS	SciFact	Touche2020	TRECCOVID	Avg
Bert-base	42.9	19.9	30.3	69.4	24.4	50.2	25.3	42.3	84.8	13.1	50.6	21.8	57.4	40.9
Bert-large	43.1	21.7	31.9	68.1	26.4	51.4	26.7	46.4	85.7	13.8	54.7	20.7	59.2	42.2
t5-v1_1-xxl	44.0	24.6	35.2	63.4	36.1	57.5	31.4	50.3	85.1	15.1	62.0	22.7	52.9	44.6
Phi-v1.5-1.3B	45.4	26.3	28.0	64.9	32.1	54.5	31.7	42.5	86.6	16.2	65.9	23.6	65.0	44.8
Phi-v2-2.7B	49.4	31.2	34.4	70.7	38.4	62.2	36.5	50.8	86.9	18.5	67.2	23.3	66.1	48.8
Gemma-2B	47.9	31.5	40.2	72.9	39.0	61.9	36.0	52.5	84.8	18.1	72.4	18.7	55.7	48.5
Gemma-7B	49.9	31.3	42.8	73.5	44.0	67.3	38.1	60.4	86.9	18.7	74.7	21.5	58.3	51.2
Llama-2-7B	48.7	31.2	44.4	76.2	42.3	68.1	36.2	57.3	86.8	18.3	73.8	19.6	47.8	50.0
Llama-2-13B	57.4	30.7	43.9	70.4	45.6	67.7	37.1	60.9	85.8	17.7	74.6	21.8	55.0	51.4
Llama-3-8B	56.1	30.8	41.6	72.7	41.7	66.0	35.2	56.4	85.8	17.8	74.0	20.6	56.9	50.4
Qwen1.5-0.5B	46.0	26.6	32.9	68.1	31.9	56.6	29.8	43.4	84.6	15.8	65.4	13.5	54.7	43.8
Qwen1.5-4B	50.2	30.5	40.5	72.9	39.4	63.7	35.4	54.3	85.3	17.5	70.8	18.3	58.6	49.0
Qwen1.5-14B	56.5	30.1	43.0	73.4	45.0	64.4	36.4	59.3	85.7	19.3	74.2	21.9	60.8	51.5
Qwen1.5-32B	57.5	31.3	44.5	75.3	47.9	68.0	37.1	59.7	86.0	18.8	75.6	24.5	60.3	52.8

Table 6: Detailed result of zero-shot retrieval generalization.

Model	256	512	1024	2048	4096	8192
BERT-large	18.0	18.1	-	-	-	-
Qwen1.5-0.5B	17.6	19.0	20.1	21.1	37.1	44.9
Qwen1.5-4B	22.8	23.9	25.4	27.1	49.1	54.9
Qwen1.5-7B	24.3	26.4	27.8	28.2	52.3	55.9
Qwen1.5-32B	26.9	28.4	28.7	30.8	54.8	59.0
Llama3-8B	28.4	29.2	29.9	30.4	53.4	57.9

Table 7: Detailed result of lengthy retrieval on narrativeqa with varying maximum input passage length.

MSMARCO Train Instructions	Given a web search query, retrieve relevant passages that answer the query. Retrieve pertinent passages from web searches to address the query. Obtain relevant excerpts from online searches to provide answers. Access passages on the web that directly respond to the search query. Find pertinent text snippets online that address the given search query. Access relevant passages from internet searches that answer the query at hand.
MSMARCO Evaluate Instructions	Locate specific passages on the web that not only answer the given query but also provide in-depth analysis and contextual information. Identify and extract relevant text from online sources that comprehensively respond to the query, including supporting evidence. Obtain detailed excerpts from reputable online searches that provide thorough answers and insights related to the query. Find relevant online snippets that respond to the given query. Retrieve and synthesize passages from websites that directly and extensively address the search query.
NQ	Given a question, retrieve Wikipedia passages that answer the question.
FiQA	Given a financial question, retrieve user replies that best answer the question.
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question.
NFCorpus	Given a question, retrieve relevant documents that best answer the question.

Figure 5: Instrctions used in instruction-based retrieval.