PromptReps: Prompting Large Language Models to Generate Dense and Sparse Representations for Zero-Shot Document Retrieval

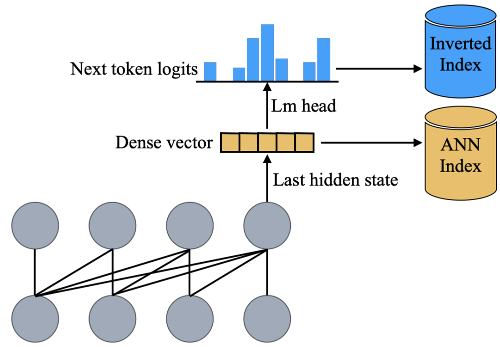
Shengyao Zhuang[1,3], **Xueguang Ma**[2], **Bevan Koopman**[1,3], **Jimmy Lin**[2], **Guido Zuccon**[3]
[1]CSIRO,
[2] University of Waterloo,
[3] The University of Queensland

Abstract

The current use of large language models (LLMs) for zero-shot document ranking follows one of two ways: 1) prompt-based re-ranking methods, which require no further training but are feasible for only re-ranking a handful of candidate documents due to the associated computational costs; and 2) unsupervised contrastive trained dense retrieval methods, which can retrieve relevant documents from the entire corpus but require a large amount of paired text data for contrastive training. In this paper, we propose PromptReps, which combines the advantages of both categories: no need for training and the ability to retrieve from the whole corpus. Our method only requires prompts to guide an LLM to generate query and document representations for effective document retrieval. Specifically, we prompt the LLMs to represent a given text using a single word, and then use the last token's hidden states and the corresponding logits associated to the prediction of the next token to construct a hybrid document retrieval system. The retrieval system harnesses both dense text embedding and sparse bag-of-words representations given by the LLM. Our experimental evaluation on the BEIR zero-shot document retrieval datasets illustrates that this simple prompt-based LLM retrieval method can achieve a similar or higher retrieval effectiveness than state-of-the-art LLM embedding methods that are trained with large amounts of unsupervised data, especially when using a larger LLM.[1]

1 Introduction

Large Language Models (LLMs) such as GPT4 and LLaMA, which are pretrained on massive corpora and finetuned to follow user instructions, have strong zero-shot natural language understanding capabilities (OpenAI, 2023; Touvron et al., 2023).



<System> *You are an AI assistant that can understand human language.*
<User> *Passage: "***[text]***". Use one most important word to represent the passage in retrieval task. Make sure your word is in lowercase.*
<Assistant> *The word is: "*

Figure 1: Overview of PromptReps, LLMs are prompted to generate dense and sparse text representation at the same time, which is then used to build search indices.

Via prompting, LLMs excel in various text generation tasks such as question answering, writing assistance, and conversational agent (Hendrycks et al., 2021; Liu et al., 2023). Inspired by the success of LLMs on natural language understanding tasks, several studies have explored the potential of using LLMs to perform unsupervised document ranking task.

One line of work focuses on directly prompting LLMs to infer document relevance to a given query (Sachan et al., 2022; Zhuang et al., 2023a; Ma et al., 2023b; Sun et al., 2023; Pradeep et al., 2023; Zhuang et al., 2023b; Qin et al., 2024). For instance, RankGPT (Sun et al., 2023) casts document re-ranking as a permutation generation task, prompting LLMs to generate re-ordered document identifiers according to the document's relevance to the query. These methods leverage LLMs for document ranking in a complete zero-shot setting where no further training is required. However, these methods can only serve as a second-stage re-ranker on a handful of candidate documents. This is because each prompt requires one full LLM inference: for example, in the case of a corpus with 1M

[1]Code for fully reproducing the results is available at https://github.com/ielab/PromptReps.

documents, a pointwise approach would require the construction of 1M prompts and thus the execution of 1M (costly) LLM inferences – making it unfeasible for an online search engine.

Another line of research leverages LLMs as a text embedding model for dense document retrieval (Lee et al., 2024; Wang et al., 2024a,b; BehnamGhader et al., 2024). For example, E5-mistral (Wang et al., 2024b) employs LLMs to create synthetic datasets of query-document pairs. These paired text data are then used to perform unsupervised contrastive training for a Mistral LLM-based dense retriever. Since the queries and documents are encoded with LLMs separately, i.e., using a bi-encoder architecture, these methods could serve as a first-stage document retriever. However, all existing LLM-based retrievers require an unsupervised contrastive training step to transform a generative LLM into a text-embedding model. Even with parameter-efficient training techniques such as LoRA (Hu et al., 2022), this extra training is still very expensive. For example, the contrastive training of E5-mistral using a large batch size (2048) and LoRA took $\approx$ 18 hours on 32 V100 GPUs (Wang et al., 2024b).

In this work, we propose a new zero-shot LLM-based document retrieval method called PromptReps. We demonstrate that LLMs can be directly prompted to produce query and document embeddings, which can serve as effective text representations for neural retrieval systems. Specifically, we prompt an LLM by asking it to use a single word to represent a query or a document. Then, we extract the last layer's hidden state of the last token in the prompt as the dense representation of the input text. Simultaneously, we utilize the logits associated with predicting the subsequent token to form a sparse representation. As illustrated in Figure 1, through a single forward pass, we generate text representations for a document that can be indexed for dense, sparse, or hybrid search architectures.

We evaluate our approach on the BEIR datasets (Thakur et al., 2021) to perform zero-shot retrieval. We utilize various open-source large language models and find that the effectiveness of zero-shot retrieval increases with the scaling of LLM size. Results show that PromptReps outperforms previous LLM-based embedding methods. Of key importance is that our method is the first LLM-based method that can effectively perform full corpus retrieval while at the same time not requiring contrastive training, demonstrating that prompt engineering for generative LLMs is capable of generating robust representations for retrieval.

2 Related Work

2.1 Supervised Neural Retrievers

Neural retrievers based on bi-encoder architecture bring significant improvement over traditional heuristic retrievers such as BM25. Dense retrievers such as DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2021), ColBERT (Khattab and Zaharia, 2020), encode text into low-dimensional dense vectors and conduct search with (approximate) nearest neighbor search. On the other hand, sparse neural retrievers such as DeepImpact (Mallia et al., 2021), uniCOIL (Lin and Ma, 2021), TILDE (Zhuang and Zuccon, 2021b,a), and SPLADE (Formal et al., 2021), encode text into high-dimensional sparse vectors as bag-of-words representations and conduct search in an inverted index. Recent work has also explored fine-tuning LLMs as dense retrievers such as RepLLaMA (Ma et al., 2023a) and LLaRA (Liao et al., 2024). A hybrid neural retrieval system refers to a system that combines the rankings provided by both dense and sparse retrievers, often resulting in an enhanced final ranking (Lin and Ma, 2021). All these retrievers are trained with supervised relevance judgment data (e.g., MS MARCO (Bajaj et al., 2018)) using contrastive learning. Our work instead focuses on building a hybrid neural retrieval system with zero-shot dense and sparse document representations without supervised contrastive learning. This is particularly valuable for applications where no relevance judgements are available — often the case in real work retrieval settings.

2.2 Unsupervised Neural Retrievers

There have also been attempts on training effective neural retrievers without relying on human relevance judgments. Methods such as Contriever (Izacard et al., 2022) and E5 (Wang et al., 2024a), train dense retriever with large-scale pseudo query-document pairs to build unsupervised (synthetic) training data. LLMs have also been adapted as unsupervised text embedding models for first-stage document retrieval. For instance, HyDE (Gao et al., 2023a) enhances query representations for an unsupervised retriever by replacing the original query with LLM-generated hypothetical documents. More recent work has focused on directly converting generative LLMs into a text-embedding models with unsupervised contrastive pre-training. These works, such as E5-Mistral-Inst (Wang et al., 2024b) and Gecko (Lee et al., 2024), use large-scale weakly supervised paired text data or LLM-generated query-document pair data to perform contrastive training on top of LLMs. LLM2Vec (BehnamGhader et al., 2024), on the other hand, conducts further masked next token prediction pre-training with bidirectional attention, and SimCSE (Gao et al., 2021) trains on raw text data to transform LLMs into text encoders. Although no labeled data is used, these methods require synthetic or unsupervised paired text data to perform contrastive pre-training. Our method instead relies solely on prompt engineering to transform LLM into a robust text encoder for document retrieval without any extra training.

2.3 Prompting LLMs for document re-ranking

Inspired by the prompt-following capacity of LLMs, recent works have explored prompting LLMs for document reranking. For instance, UPR (Sachan et al., 2022) ranks documents pointwise by prompting the LLM to generate a relevant query for a given document and rank documents based on the likelihood of generating the query. RankGPT (Sun et al., 2023) and LRL (Ma et al., 2023b) propose to re-rank a list of documents at once and generate permutations for the reordered list. Pairwise (Qin et al., 2024) and Setwise (Zhuang et al., 2023b) prompting methods have also been explored to improve effectiveness and efficiency in the LLM re-ranking pipeline. These methods are only feasible for re-ranking a handful of candidate documents, thus limited to second-stage document re-ranking. In contrast, our approach utilizes prompts to construct the first-stage retrievers.

2.4 Prompting LLM for sentence embeddings

The methods most similar to ours are work on prompting LLMs to generate sentence embeddings for semantic textual similarity (STS) tasks (Jiang et al., 2023b; Lei et al., 2024; Zhang et al., 2024). These works developed an Explicit One-word Limitation (EOL) prompt, which also instructs LLMs to represent a sentence with one word. However, these works only evaluate such prompts on STS datasets, and their effectiveness on information retrieval datasets with large document corpora is unknown. Additionally, these methods only represent texts with dense embeddings from the hidden states; our method instead generates sparse text representations simultaneously to build a hybrid retrieval system. In fact, our results show that dense embeddings alone perform poorly for document retrieval tasks with some LLMs, but sparse representations are much more robust, and the best retrieval effectiveness is achieved with the hybrid retrieval system with scaled model size.

3 PromptReps

Unlike previous works that leverage LLMs for document ranking, which are limited to document reranking tasks with prompts or rely on contrastive learning to transform a generative LLM into an embedding model for document retrieval, in this work, we aim to directly prompt LLMs to generate both dense embedding representations and sparse bag-of-words representations for document retrieval without any form of extra training effort. To achieve this aim, we devise the following prompt as the input text for LLMs:

> **<System>** *You are an AI assistant that can understand human language.*
> **<User>** *Passage "***[text]***". Use one most important word to represent the passage in retrieval task. Make sure your word is in lowercase.*
> **<Assistant>** *The word is: "*

where <System> <User> and <Assistant> are LLM pre-defined conversational prefix tokens and [text] is the placeholder for passage text. When using this prompt for text generation, the language model needs to find a single word in its token vocabulary that can best represent the given passage to generate. However, since there could be multiple words to represent the passage, there might be multiple tokens in the vocabulary that have a high probability of being sampled by the language model. Such a distribution over the vocabulary, which often refers to "logits", could potentially provide a good representation of the given passage. In addition, since the logits are computed by the last layer hidden state[2] of the last input token ('"'), which is a dense vector embedding, it can also serve as a dense representation of the passage.

Based on the above intuition, we develop a sparse + dense hybrid document retrieval system

[2]Often through dot product between last hidden state with all token embeddings.

by utilizing both the next token logits and the last layer hidden states outputted by the LLM with our designed prompt.

Specifically, during the document indexing phase, we pass all the documents with our prompt into the LLM to get output hidden states and logits. To build a sparse retrieval pipeline with logits, we first need to sparsify the logits representation to be able to perform efficient sparse retrieval. This is because logits originally had values for all tokens in the vocabulary, essentially forming dense vectors with dimensions equal to the vocabulary size. To sparsify the logit representations for sparse retrieval, we perform following steps:

1. Lowercase the input document text to align with the phrase "Make sure your word is in lowercase." in the prompt since this phrase skewed the sampling distribution towards lowercase tokens (a "sparser" distribution). We then utilize the NLTK toolkit (Bird and Loper, 2004) to extract all words in the document, filtering out standard English stopwords and punctuation.

2. Next, we use the LLM's tokenizer to tokenize each extracted word and obtain their token IDs[3]. We retain only the values corresponding to the obtained token IDs in the logits and set the rest of the dimensions to zero, thereby considering only tokens present in the documents, thus enabling exact term matching in retrieval.

3. Next, we follow the SPLADE recipe (Formal et al., 2021), using the ReLU function to remove dimensions with negative values and applying log-saturation to the logits to prevent certain tokens from dominating. To further enhance the sparsity of logits, we only keep tokens within the top 128 values if the logits had more than 128 non-zero values after the previous steps.

4. Finally, the logits were quantized by simply multiplying the original values by 100 and taking the integer operation on that, and the obtained values represent the weights of corresponding tokens.

[3]Note that many words may be split into sub-tokens, resulting in multiple token IDs, all of which are considered in the logits

With these adjustments, the logits representations of documents are heavily sparsified, allowing for efficient sparse retrieval with a standard inverted index.

For dense retrieval, we directly use the hidden states as the embedding of the documents. For indexing these embeddings, we simply normalize all the embeddings and add them into an Approximate Nearest search (ANN) vector index.

At query time, we process the queries exactly the same as the documents, with the only exception being that the term "passage" in the prompt is replaced with "query". The dense representation of the query is utilized for semantic search via the ANN index, while the sparse representation of the query is employed for exact term matching via the inverted index. Following previous work (Wang et al., 2021), we compute the final document scores by applying min-max normalization to both dense and sparse document scores. These normalized scores are then linearly interpolated with equal weights to produce the final document scores.

4 Experimental setup

Dataset and evaluation: We evaluate the document ranking effectiveness of both baseline methods and our proposed PromptReps using the BEIR benchmark (Thakur et al., 2021). This benchmark encompasses various IR tasks, providing a heterogeneous evaluation environment. We report nDCG@10 scores, which is the official evaluation metric employed by BEIR.

Baselines: We compare PromptReps with strong unsupervised first-stage retrievers including BM25, a classic term frequency-based sparse retrieval method, and E5-PT$_{large}$ (Wang et al., 2024a), a state-of-the-art BERT-based (Devlin et al., 2019) dense embedding method trained on 1.3 billion carefully crafted unsupervised text pairs. LLM2Vec (BehnamGhader et al., 2024), an LLM-based dense embedding method trained with bi-directional attention, masked next token prediction, and SimCSE (Gao et al., 2021) on the Wikipedia corpus.

Implementation of PromptReps: PromptReps is implemented using three base LLMs: Mistral-7b-Instruct-v0.2[4] (Jiang et al., 2023a), Llama3-

[4]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

Table 1: nDCG@10 scores of BEIR 13 publicly available datasets. The best scores of methods without interpolating with BM25 are highlighted in bold. Scores of E5-PT$_{large}$ and LLM2Vec are taken directly from the original papers.

		Contrastive pre-training		PromptReps(ours)													
LLM	-	BERT-330M	Mistral-7B	Mistral-7B-Instruct			Phi3-mini-3.8B-Instruct			Llama3-8B-Instruct			Llama3-70B-Instruct				
Dataset	BM25	E5-PT$_{large}$	LLM2Vec	Dense	Sparse	Hybrid	Dense	Sparse	Hybrid	Dense	Sparse	Hybrid	Dense	Sparse	Hybrid	+BM25
arguana	39.70	44.4	**51.00**	21.02	32.41	32.81	38.00	21.51	35.68	32.89	24.25	35.47	37.38	26.15	38.29	42.28
climate-fever	16.51	15.7	**22.97**	5.31	11.17	10.57	18.99	11.39	21.12	23.47	9.52	22.20	21.21	12.01	22.56	22.96
dbpedia	31.80	**37.1**	25.48	8.23	22.80	17.87	26.09	24.72	32.21	30.95	28.11	36.30	31.88	28.69	36.59	40.48
fever	65.13	68.6	45.11	20.03	50.07	47.23	44.18	51.73	64.92	59.43	51.60	**71.61**	57.62	54.77	71.52	76.37
fiqa	23.61	**43.2**	27.24	8.31	18.63	15.75	21.08	20.40	27.01	24.79	19.66	29.54	29.99	22.22	33.34	33.20
hotpotqa	**63.30**	52.2	54.54	2.98	32.15	20.62	25.15	42.75	46.62	28.07	45.82	51.27	30.94	44.91	50.98	66.42
nfcorpus	32.18	33.7	27.16	14.41	27.81	23.55	27.80	29.38	33.25	26.74	27.48	31.36	31.93	29.35	**34.03**	36.19
nq	30.55	41.7	34.16	7.12	19.93	16.62	26.47	25.57	34.78	35.02	28.34	41.62	43.38	30.76	**45.98**	44.22
quora	78.86	**86.1**	84.40	53.38	61.95	66.65	80.71	67.79	82.11	77.47	69.02	82.47	75.27	71.57	81.81	84.84
scidocs	14.90	**21.8**	15.35	6.54	11.42	10.28	13.35	12.24	15.58	16.77	11.52	16.93	18.31	13.22	17.96	17.73
scifact	67.89	**72.3**	68.67	22.71	55.27	48.15	51.69	59.94	64.86	54.26	59.06	65.61	58.92	61.53	67.41	72.09
trec-covid	59.47	61.8	55.66	34.94	62.03	53.79	51.16	56.53	62.48	63.54	55.19	68.54	71.47	63.19	**77.13**	77.94
touche	**44.22**	19.8	6.54	9.59	19.47	18.36	12.75	19.17	19.58	14.92	16.46	19.24	18.84	20.12	22.85	36.21
avg	43.70	44.61	39.86	16.51	32.70	29.40	33.65	34.09	41.55	37.48	34.31	44.01	40.55	36.80	**46.19**	50.07

8B-Instruct[5], Phi-3-mini-4k-instruct[6] (Abdin et al., 2024), and Llama3-70B-Instruct[7] (AI@Meta, 2024). Dense and sparse document and query encoding are implemented using the Huggingface Transformers library (Wolf et al., 2020) and the Tevatron toolkit (Gao et al., 2023b). Faiss library (Douze et al., 2024) is used to build the ANN index with cosine similarity as the embedding distance metric, and Pyserini (Lin et al., 2021) is utilized to construct the inverted index for sparse retrieval. For the dense and sparse ranking hybrid, the Ranx library (Bassani and Romelli, 2022) is employed. In our experiments, we report dense only, sparse only, and the full hybrid results. In addition, we also report the results of further interpolating BM25 scores with hybrid scores for PromptReps based on Llama3-70B-Instruct.

5 Results

We present our results in Table 1. The first observation highlights that BM25 is a very strong zero-shot retrieval method, capable of outperforming LLM2Vec, based on the Mistral-7B LLM, across numerous datasets, achieving a higher average nDCG@10 score. This outcome implies that even with a large-size LLM, bi-directional attention enabled, additional pre-training, and SimCSE-based unsupervised contrastive training, there remains a gap in transforming a decoder-only LLM into a robust retrieval method.

On the other hand, E5-PT$_{large}$, based on the BERT-large model, is the first method that can outperform BM25 without any supervised training data. However, it has been trained on a massive, carefully mined text pair dataset with a large batch size, which may require more data-collecting efforts and computational resources than LLM2Vec.

For our proposed PromptReps, when using Mistral-7B-Instruct LLM, the dense-only retriever performs poorly. However, although still worse than LLM2Vec, the sparse-only retriever performs much better than dense. This finding suggests that, with Mistral-7B-Instruct as the base LLM, using the logits generated by our prompt produces better document representations than hidden states for document retrieval. The hybrid retrieval performs worse than sparse-only in this case due to the poor performance of dense retrieval.

Surprisingly, when implementing PromptReps with Phi-3-mini-4k-instruct achieved much higher dense-only retrieval effectiveness, closing to its sparse retrieval effectiveness and leading to an impressive hybrid retrieval average score that surpassed LLM2Vec based on Mistral-7B, even though it is only half the size.

Interestingly, when implementing PromptReps with the state-of-the-art Llama3-8B-Instruct LLM, the dense-only retrieval further improved and achieved a higher score than that of sparse. The sparse-only retrieval effectiveness also improved, but the improvement was much more marginal compared to the improvement on dense. The hybrid of dense and sparse brings notable retrieval effectiveness improvement, surpassing BM25 and approaching the state-of-the-art E5-PT$_{large}$. Notably, this is achieved without any form of extra training but solely relying on prompts.

The scaling law of LLM (Kaplan et al., 2020) is also applied here. When changing Llama3-8B-Instruction to Llama3-70B-Instruction, the retrieval effectiveness of our PromptReps further improved,

[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[6] https://huggingface.co/microsoft/Phi-3-mini-4k-instruct
[7] https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct

with the dense-only approach surpassing LLM2Vec and the hybrid approach surpassing E5-PT$_{large}$. When further interpolating the hybrid ranking with BM25 ranking, the average nDCG@10 achieved a remarkable score of 50.07. Our results demonstrate that it is possible to build a very strong hybrid retrieval system without the need for any unsupervised or supervised training.

A note on query latency. The best performance of our PromptReps is achieved in the hybrid retrieval setting. Compared to previous works which are dense retrieval only, our method seems to have a higher query latency. However, PromptReps actually only requires a little query latency overhead if implementing dense and sparse search in parallel. The cost of obtaining both dense and sparse representations of our method only requires a single LLM forward inference; the only extra computation is the dot product of the dense vector with the token embeddings, which is very fast on GPU. For document search, since we heavily sparsified the sparse representation, in our experiments, our sparse retrieval is much faster than BM25, and the bottleneck is the dense retrieval. Since the dense and sparse search could be run in parallel and the hybrid operation is just a linear interpolation of both rankings (very fast on CPU), the query latency will depend on the dense retrieval latency, thus very close to the previous methods.

6 Discussion

Our work demonstrates that prompt engineering for generative LLMs is not only a powerful method for natural language generation tasks but also capable of generating robust dense (embedding) and sparse (bag of words) representations for input texts. We believe this important finding could lead to a series of new research directions.

For example, techniques like few-shot in-context learning (Brown et al., 2020), chain-of-thought prompting (Wei et al., 2022), and auto-prompt optimization methods (Yang et al., 2024; Fernando et al., 2023), which have proven to be very effective in text-generation tasks, could potentially be leveraged here to enhance embedding generation.

Moreover, it has been shown that the instruction-following ability of LLMs could be transferred to embedding models with synthetic instruction fine-tuning data (Wang et al., 2024b). In our work, we always keep the instruction prompt consistent across different information retrieval tasks, which could be sub-optimal. It would be very interesting to investigate how to customize instructions for PromptReps to generate embeddings specific to different domains, tasks, or even to multi-lingual and cross-lingual retrieval settings.

Finally, our prompting method could be seen as a simple approach to obtaining a better initialization of LLM-based embedding models, which is much more cost-effective than methods requiring further pre-training (BehnamGhader et al., 2024; Li et al., 2023). All the previous contrastive pre-training with paired text data and synthetically generated data could be applied on top of our method and could potentially yield improved LLM-based embedding models. We leave these aspects for future work.

7 Conclusion

In this paper, we introduced PromptReps, a simple yet effective method that prompts LLMs to generate dense and sparse representations for zero-shot document retrieval without any further unsupervised or supervised training. Our work reveals that modern LLMs are effective text encoders by themselves, and prompt engineering is sufficient to stimulate their text encoding ability.

References

Marah Abdin et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

AI@Meta. 2024. Llama 3 model card.

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

Elias Bassani and Luca Romelli. 2022. ranx.fuse: A python library for metasearch. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4808–4812, New York, NY, USA. Association for Computing Machinery.

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

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *Preprint*, arXiv:2401.08281.

Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *Preprint*, arXiv:2309.16797.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA. Association for Computing Machinery.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023b. Tevatron: An efficient and flexible toolkit for neural retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 3120–3124, New York, NY, USA. Association for Computing Machinery.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023b. Scaling sentence embeddings with large language models. *Preprint*, arXiv:2307.16645.

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. *Preprint*, arXiv:2001.08361.

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

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Jinhyuk Lee, Zhuyun Dai, Xiaoqi Ren, Blair Chen, Daniel Cer, Jeremy R. Cole, Kai Hui, Michael Boratko, Rajvi Kapadia, Wen Ding, Yi Luan, Sai Meher Karthik Duddu, Gustavo Hernandez Abrego, Weiqiang Shi, Nithi Gupta, Aditya Kusupati, Prateek Jain, Siddhartha Reddy Jonnalagadda, Ming-Wei Chang, and Iftekhar Naim. 2024. Gecko: Versatile text embeddings distilled from large language models. *Preprint*, arXiv:2403.20327.

Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. 2024. Meta-task prompting elicits embedding from large language models. *Preprint*, arXiv:2402.18458.

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

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. Llara: Large language-recommendation assistant. *Preprint*, arXiv:2312.02445.

Jimmy Lin and Xueguang Ma. 2021. A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques. *Preprint*, arXiv:2106.14807.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, Zihao Wu, Lin Zhao, Dajiang Zhu, Xiang Li, Ning Qiang, Dingang Shen, Tianming Liu, and Bao Ge. 2023. Summary of chatgpt-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017.

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

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023b. Zero-shot listwise document reranking with a large language model. *Preprint*, arXiv:2305.02156.

Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonellotto. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1723–1727, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2023. GPT-4 technical report. *arXiv:2303.08774*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *Preprint*, arXiv:2309.15088.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. *Preprint*, arXiv:2306.17563.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024a. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024b. Improving text embeddings with large language models. *Preprint*, arXiv:2401.00368.

Shuai Wang, Shengyao Zhuang, and Guido Zuccon. 2021. Bert-based dense retrievers require interpolation with bm25 for effective passage retrieval. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 317–324, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Bowen Zhang, Kehua Chang, and Chunping Li. 2024. Simple techniques for enhancing sentence embeddings in generative language models. *Preprint*, arXiv:2404.03921.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023a. Open-source large language models are strong zero-shot query likelihood models for document ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8807–8817, Singapore. Association for Computational Linguistics.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2023b. A setwise approach for effective and highly efficient zero-shot ranking with large language models. *Preprint*, arXiv:2310.09497.

Shengyao Zhuang and Guido Zuccon. 2021a. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *Preprint*, arXiv:2108.08513.

Shengyao Zhuang and Guido Zuccon. 2021b. Tilde: Term independent likelihood model for passage re-ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1483–1492, New York, NY, USA. Association for Computing Machinery.