InstUPR : Instruction-based Unsupervised Passage Reranking with Large Language Models

Anonymous ACL submission

Abstract

This paper introduces InstUPR , a novel unsupervised passage reranking method based on large language models (LLMs). Different from existing approaches that rely on extensive training with query-document pairs or retrieval-specific instructions, our method leverages the instruction-following capabilities of instruction-tuned LLMs for passage reranking without any additional fine-tuning. To achieve this, we introduce a soft score aggregation technique and employ pairwise reranking for unsupervised passage reranking. Experiments on the BEIR benchmark demonstrate that InstUPR outperforms unsupervised baselines as well as an instruction-tuned reranker, highlighting its effectiveness and superiority.

1 Introduction

Information retrieval (IR) involves the retrieval of relevant information from a large collection of data, such as web pages or documents, in response to a user's query. Recently, deep learning methods like dense passage retriever (DPR) (Karpukhin et al., 2020) have gained significant interest due to their superior performance compared to sparse retrieval methods such as BM25. However, it is crucial for initial retrievers to be lightweight to handle a large set of retrieval targets. Therefore, passage reranking plays a crucial role in the process by following the initial retrievers and ranking the retrieved passages based on their relevance to the query. This enables the use of computationally intensive models, thereby enhancing retrieval accuracy.

Large language models (LLMs) have demonstrated strong zero-shot capabilities across various natural language tasks (Brown et al., 2020; Kojima et al., 2022). Specifically, models fine-tuned on natural language instructions have shown remarkable performance in comprehending complex instructions (Wei et al., 2021). Previous work has explored the use of LLMs for passage reranking by fine-tuning them on extensive retrieval supervision (Nogueira et al., 2020; Asai et al., 2022). Another line of investigation involves unsupervised passage reranking using LLMs (Sachan et al., 2022; Sun et al., 2023). However, these unsupervised methods often lack guidance in understanding the relevance of retrieved passages.

This paper introduces InstUPR , an instruction-based unsupervised passage reranking method that leverages the instruction-following capabilities of LLMs for reranking *without* the need for labeled relevance information and additional fine-tuning. We employ an instruction-tuned LLM to generate a relevance score for each query-passage pair. Additionally, we propose a soft relevance score aggregation technique that combines the LLM's predicted distribution over possible scores, resulting in robust estimation. We evaluate our method on common evaluation benchmarks, including TREC DL19 (Craswell et al., 2020), DL20 (Craswell et al., 2021), and BEIR (Thakur et al., 2021). Experimental results demonstrate that our InstUPR outperforms unsupervised baselines like UPR and an instruction-tuned reranker. Furthermore, our proposed soft aggregation method significantly contributes to these improvements.

Our contribution can be summarized as 3-fold:

- We propose InstUPR , which leverages the instruction-following capabilities of LLMs for unsupervised passage reranking.
- We introduce soft relevance score aggregation to enhance reranking performance.
- We propose both pointwise and pairwise reranking schemes and demonstrate their effectiveness compared with unsupervised baselines and models specifically fine-tuned on retrieval datasets.

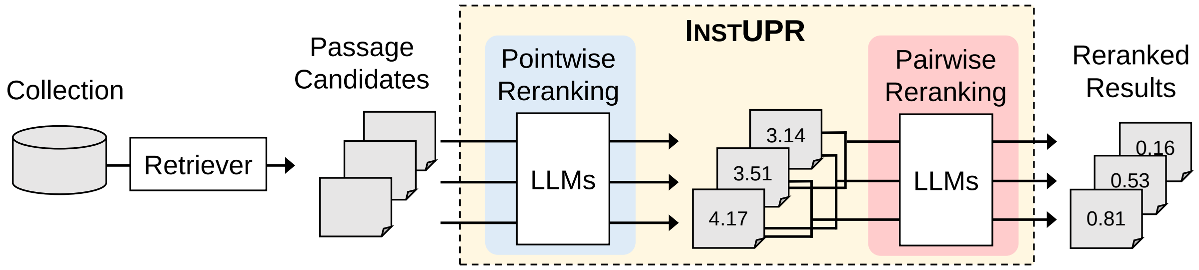

Figure 1: Illustration of our proposed INSTUPR framework, which includes pointwise reranking and pairwise reranking modules for fine-grained estimation.

2 Related Work

Information Retrieval In recent years, deep learning-based retrieval models have achieved remarkable performance across various information retrieval tasks. The dense passage retriever (DPR) framework, which encodes documents and queries into dense representations, has emerged as a popular approach for dense retrieval (Karpukhin et al., 2020). With the advent of large language models (LLMs), numerous methods have leveraged these models for dense retrieval. GTR (Ni et al., 2022) utilizes LLM encoders for dense retrieval and demonstrates performance improvements with increased model size. Promptagator (Dai et al., 2023) and InPars (Bonifacio et al., 2022) propose the use of LLMs to generate synthetic query-document pairs, which are then employed for training dense retrievers. Our work is orthogonal to these methods, as we focus on utilizing LLMs for second-stage passage reranking.

Passage Reranking Passage reranking typically serves as a second-stage component following large-scale retrieval. Several studies have proposed deep reranking models that encode query-document pairs to predict relevance scores (Nogueira and Cho, 2019). Nogueira et al. (2020) introduced a generation-based method for passage reranking by fine-tuning LLMs on MS-MARCO(Bajaj et al., 2016), a large-scale retrieval dataset with relevance annotations. Their model, MonoT5, generates the word `true` for relevant pairs and `false` for irrelevant pairs. Similarly, our method also adopts a generation-based approach. The main difference is that our method does not require relevance annotations nor fine-tuning; instead, we leverage the instruction-following capabilities of LLMs to enable unsupervised estimation. TART (Asai et al., 2022) fine-tunes LLMs on extensive retrieval supervision from various tasks with instructions. Our method differs from TART in that we do not require any retrieval supervision and employ a generation-based approach in an unsupervised fashion.

Another research line is unsupervised passage reranking with LLMs, which eliminates the need for retrieval supervision. UPR (Sachan et al., 2022) is the pioneering attempt at unsupervised passage reranking, proposing to rerank passages by estimating the conditional likelihood of generating the query given the passage using LLMs. UPR has shown promising results, but it employs an indirect measure that may not be optimal for measuring the relevance of retrieved passages. In contrast, our INSTUPR leverages the instruction-following capabilities of LLMs while requiring no retrieval supervision. Through extensive experiments, we demonstrate that INSTUPR outperforms UPR on most datasets, highlighting its effectiveness. Concurrent to our work, Sun et al. (2023) and Ma et al. (2023) both proposed to perform listwise passage reranking by prompting ChatGPT, which is a black-box commercial system [1]. Our work focuses on pointwise and pairwise reranking, and employs an open-sourced LLM with well-documented data sources to facilitate scientific understanding of our method.

3 Our Method

The task of passage reranking involves assigning a relevance score to each document in a set of retrieved candidates given a query. Formally, given a query q and a set of retrieved passages $D = d_1, d_2, \cdots, d_k$, a reranker aims to assign a relevance score to each query-passage pair as $s(q, d_i)$. These relevance scores are then used to rerank the passage candidates. Figure 1 illustrates the proposed reranking framework.

[1] https://chat.openai.com/

3.1 InstUPR: Instruction-based Unsupervised Passage Reranking

Our method, InstUPR, leverages the instruction-following capabilities of LLMs to enhance the performance of passage reranking. We prompt the LLMs with task-specific instructions that instruct them to directly generate a relevance score for each query-passage pair (q, d_i) and rerank the passage candidates based on their relevance scores. In this paper, we instruct the LLMs to predict a relevance score from 1 to 5 using the Likert scale. For parsing convenience, we instruct the LLMs to generate only a single token from the options, which in our case are $1, 2, 3, 4, 5$. An example of the instruction template is shown in Figure 2a.

3.2 Soft Relevance Score Aggregation

Generating a single token as the relevance score introduces several issues (Liu et al., 2023). First, it results in discrete scores that lead to many ties, which is suboptimal for reranking. Second, we observe that the generated scores tend to be very similar for the same task, such as the LLM frequently outputting a score of 3 for the majority of the passages. To address these issues, we propose *Soft Relevance Score Aggregation*. Instead of using the generated token directly, we compute a weighted sum of the options using their probabilities as weights. Specifically, the soft relevance score of a query-passage pair $s_1(q, d_i)$ can be calculated as:

$$s_1(q, d_i) = \sum_{n=1}^{5} n \cdot p(n \mid q, d_i),$$

where $p(n \mid q, d_i)$ is the probability of predicting a score of n by the LLM. This score can also be interpreted as the expected value of the score predicted by the LLM.

3.3 Pairwise Reranking

Pairwise reranking has been demonstrated to outperform pointwise reranking while being more computationally expensive (Nogueira et al., 2019; Pradeep et al., 2021). Given a query q and two passages d_i and d_j, we instruct the LLM to select the passage that is more relevant to the query and assign the probability of selecting each passage as the score. The final score of a passage d_i, denoted as $s_2(q, d_i)$, is then re-estimated as the sum of its scores against all other passage candidates:

$$s_2(q, d_i) = \sum_{i \neq j} p(i \mid q, d_i, d_j),$$

where $p(i \mid q, d_i, d_j)$ is the probability predicted by the LLM of d_i being more relevant to the query q than d_j. It is important to note that the ordering of passages affects the scores, i.e., $p(i \mid q, d_i, d_j) \neq p(i \mid q, d_j, d_i)$. Therefore, we evaluate all $(k^2 - k)$ pairs to obtain the pairwise rankings for robustness. An instruction template is shown in Figure 2b.

Rate the relevance of the query and the context with a score from 1 to 5, where 1 means “completely irrelevant” and 5 means “completely relevant”.
Query: {query}
Context: {context}
Score:

3

(a) Instrcution for pointwise reranking.

Which context is more relevant to the query (A or B)?
Query: {query}
Context A: {contextA}
Context B: {contextB}

A

(b) Instruction for pairwise reranking.

Figure 2: The instruction templates for reranking in InstUPR.

4 Experiments

4.1 Setup

To evaluate the effectiveness of our proposed InstUPR, we conduct experiments on TREC DL19 (Craswell et al., 2020), DL20 (Craswell et al., 2021), and BEIR (Thakur et al., 2021), which consists of various tasks for zero-shot retrieval and reranking. Following previous work, we employ BM25 as the base retrieval method and retrieve the top-100 passages for reranking (Rosa et al., 2022). For our experiments, we utilize `flan-t5-xl` (Chung et al., 2022) as our LLM to ensure that it has not been pretrained on our specific datasets. We report **NDCG@10**, which is the standard metric for evaluating retrieval performance. Additional details can be found in Appendix A.

4.2 Baseline Systems

- **TART-Rerank** (Asai et al., 2022) is a state-of-the-art reranker that is fine-tuned on a collection of retrieval datasets with instructions.
- **MonoT5-3B** (Nogueira and Cho, 2019) is a reranker that is fine-tuned on MS MARCO for predicting whether the passage is relevant to the query.
- **UPR** (Sachan et al., 2022) is an unsupervised reranking method that reranks passages by

			Supervised		Unsupervised		
		BM25	**TART-Rerank**	**MonoT5-3B**	**UPR**	**InstUPR$_{point}$**	**InstUPR$_{+pair}$**
TREC	DL19	50.58	67.43	**71.83**	54.51	61.61	70.53
	DL20	47.96	59.19	**68.89**	55.91	61.88	68.55
BEIR	TREC-COVID	59.47	74.20	80.71	69.25	73.04	**81.33**
	BioASQ	52.25	56.20	57.50	56.59	55.17	**59.25**
	NFCorpus	32.18	33.70	**38.97**	33.78	35.24	37.10
	FiQA	23.61	35.70	**45.99**	37.19	39.76	41.24
	Signal-1M	**33.04**	28.38	32.55	31.78	32.58	31.26
	TREC-News	39.52	42.63	48.49	36.06	46.12	**50.37**
	Robust04	40.70	50.63	56.71	44.40	54.03	**60.23**
	Touche-2020	**44.22**	28.33	32.41	21.07	28.98	34.22
	DBPedia	31.80	42.53	**44.45**	30.72	42.43	43.53
	SCIDOCS	14.90	17.34	19.00	15.88	18.97	**20.68**
	Climate-FEVER	16.51	27.21	**27.33**	18.25	26.18	25.81
	SciFact	64.76	75.19	**76.57**	73.09	71.28	74.96
	Average (BEIR)	38.01	42.67	**46.65**	39.01	43.66	46.63

Table 1: Reranking performance (NDCG@10) of both supervised and unsupervised methods (%); the best scores are in **bold**, and the second best scores are underlined.

their conditional probabilities of generating the query. We use `flan-t5-xl` for UPR for a fair comparison.

	DL19	DL20	BEIR
InstUPR$_{point}$	**61.61**	**61.88**	**43.66**
- soft aggregation	57.08	58.13	37.13
- Likert scale	58.27	57.46	38.36

Table 2: Results of ablation study (%).

4.3 Main Results

The experimental results are presented in Table 1. In comparison to the unsupervised baseline UPR, our InstUPR$_{point}$ outperforms UPR in 12 out of the 14 tasks, exhibiting an average relative improvement of over 10%. Furthermore, InstUPR$_{point}$ outperforms TART-Rerank in 8 tasks, despite not being trained with any retrieval supervision. It highlights the effectiveness of our proposed instruction-based reranking method, which directly leverages the instruction-following capabilities of LLMs.

With the inclusion of our proposed unsupervised pairwise reranking (InstUPR$_{+pair}$), we achieve the best performance in 5 tasks and the second-best performance in 6 tasks. Remarkably, InstUPR$_{+pair}$ achieves comparable performance to the state-of-the-art reranker MonoT5-3B while being an unsupervised method, demonstrating its practical value for real-world applications.

4.4 Ablation Study

To validate the effectiveness of individual components, we conduct an ablation study presented in Table 2. Removing the soft score aggregation component leads to significant degradation in all tasks, highlighting the importance of our proposed soft score aggregation for robust estimation. We also examine the impact of removing the Likert scale and directly asking the LLM whether the passage is relevant to the query, using the probability of generating “`yes`” as the relevance score. The results demonstrate a substantial drop after removing the Likert-based scores, showing the effectiveness of our proposed scoring method.

5 Conclusion

In this paper, we propose InstUPR, an instruction-based unsupervised passage reranking method. We leverage the instruction-following capabilities of LLMs for passage reranking and propose soft score aggregation and pairwise reranking to further improve the performance. Experimental results show that InstUPRoutperforms previous unsupervised methods and achieves comparable performance to the state-of-the-art method, demonstrating the great potential of leveraging LLMs for information retrieval tasks. We hope our work can draw attention to exploring the application of LLMs to information retrieval studies. Future work could explore how the scale of LLMs affects reranking performance and efficient pairwise reranking techniques.

Limitations

While our proposed method demonstrates impressive performance, it is important to acknowledge certain limitations. First, the pairwise reranking approach we employ incurs high computational costs, making it challenging to scale up to scenarios involving hundreds of passage candidates. Future research could focus on exploring more efficient pairwise reranking techniques to address this limitation. Second, our experiments are conducted using a single large language model (LLM), and it is possible that different LLMs may exhibit varying behaviors and performances. To address this, further investigation is needed to assess the generalization capabilities across diverse LLMs.

Ethics Statement

In this study, we utilize an instruction-following LLM that has been pre-trained on extensive text data and subsequent fine-tuning with instructions. It is important to recognize that LLMs have the potential to exhibit biased and offensive behavior, which can impact the quality and veracity of the reranking results. Careful attention should be given to mitigating bias and ensuring ethical considerations are taken into account when deploying such models in real-world applications.

References

Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. 2022. Task-aware retrieval with instructions. *arXiv preprint arXiv:2211.09260*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Unsupervised dataset generation for information retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2387–2392.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. corr abs/2102.07662 (2021). *arXiv preprint arXiv:2102.07662*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*.

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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *ICML 2022 Workshop on Knowledge Retrieval and Language Models*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

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

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *arXiv preprint arXiv:2101.05667*.

Guilherme Moraes Rosa, Luiz Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2022. No parameter left behind: How distillation and model size affect zero-shot retrieval. *arXiv preprint arXiv:2206.02873*.

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, Pengjie Ren, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agent. *arXiv preprint arXiv:2304.09542*.

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)*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

A Additional Details

A.1 Dataset

For fair comparisons, we exclude the datasets NaturalQuestions, HotpotQA, Quora, and FEVER from BEIR as they are part of either our LLM's training set or the baselines' training set. Additionally, we exclude CQADupStack due to its evaluation complexity and its large number of queries. Also, we exclude Arguana since it is a passage-level retrieval task.

A.2 Implementation Details

For pairwise reranking, given the top-k retrieval results, we evaluate $(k^2 - k)$ pairs to obtain the pairwise scores. To reduce computations, we reduce k from 100 to 40 for smaller datasets and 15 for larger datasets. All experiments are conducted on 2xNVIDIA V100 GPUs. Future work could explore efficient pairwise reranking algorithms, such as applying sorting algorithms to pairwise reranking.