A Two-Stage Adaptation of Large Language Models for Text Ranking

Anonymous ACL submission

Abstract

Text ranking is a critical task in information retrieval. Recent advances in pre-trained language models (PLMs), especially large language models (LLMs), present new opportunities for applying them to text ranking. While supervised fine-tuning (SFT) with ranking data has been widely explored to better align PLMs with text ranking goals, previous studies have focused primarily on encoder-only and encoder-decoder PLMs. Research on leveraging decoder-only LLMs for text ranking remains scarce. An exception to this is RankLLaMA (Ma et al., 2023a), which uses direct SFT to explore LLaMA's potential for text ranking. In this work, we propose a two-stage progressive paradigm to better adapt LLMs to text ranking. First, we conduct continual pre-training (CPT) of LLMs on a large weakly-supervised corpus. Second, we perform SFT, and propose an improved optimization strategy building upon RankLLaMA. Our experimental results on multiple benchmarks show that our approach outperforms previous methods in both in-domain and out-domain scenarios.

1 Introduction

Text ranking is to order a set of candidate documents by their relevance to a given query. This process is often the second step in information retrieval, following the initial collection of candidate documents from a large corpus by a fast retriever (Robertson and Zaragoza, 2009)[1]. Early work relied primarily on the handcrafted numerical features based on query-document pairs (Chapelle and Chang, 2011). Recent advances in PLMs such as BERT (Kenton and Toutanova, 2019), along with large-scale annotated datasets like MS MARCO (Nguyen et al., 2016), have significantly improved model performance in text ranking.

[1] As the candidate set is usually small, while the first step focuses on efficiently collecting candidate documents, text ranking tends to prioritize performance over efficiency.

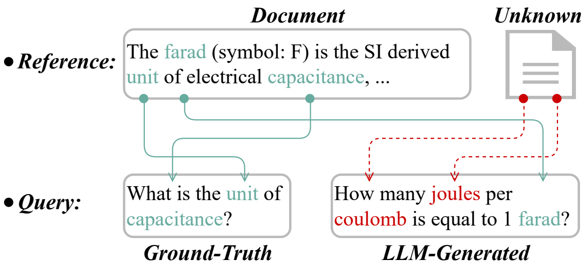

Figure 1: Misalignment between LLMs (LLaMA) and text ranking objectives: Sachan et al. (2022) measures relevance using the probability of generating a query given the document. Unlike ground-truth queries, LLM-generated queries could contain document-irrelevant terms. Such misalignment would lead to suboptimal ranking performance with out-of-the-box LLMs.

LLMs, such as LLaMA (Touvron et al., 2023a) and GPT4 (OpenAI, 2023), have brought a paradigm shift in natural language processing through their impressive performance on various tasks. This has driven growing interest in applying LLMs to text ranking (Liang et al., 2022). Recent works have explored prompt learning, as well as pointwise (Sachan et al., 2022), pairwise (Qin et al., 2023) and listwise (Sun et al., 2023b) text ranking schemas to enable out-of-the-box LLMs to perform unsupervised ranking. With the help of LLMs, substantial improvements over the BERT-style PLM counterparts have been achieved.

However, a misalignment persists between the LLM pre-training and the text ranking, as shown in Figure 1. Several studies address this through SFT of encoder-only (Nogueira et al., 2019) and encoder-decoder (Zhuang et al., 2023b) models. Yet rare work has targeted decoder-only LLMs. RankLLaMA (Ma et al., 2023a) might be the only exceptional work exploring SFT on decoder-only LLMs. While RankLLaMA shows some gains, its achievements still lag behind those of previous SFT studies, and the observation could be even more serious when tested on the out-domain scenario.

In this work, we propose a two-stage training framework to adapt decoder-only LLMs to text ranking progressively: (1) CPT followed by (2) SFT. Given the broad definition of text relevance, e.g., reasoning and semantic similarity, we first exploit a CPT stage to teach LLMs various cases of relevance. This helps the second-stage SFT more readily and accurately align LLMs with text ranking objectives. During CPT, we construct a large-scale weakly-supervised text-pair dataset, and then perform the next-token prediction task (NTP) (Radford et al., 2018) on it. For SFT, we introduce a new optimization objective different from RankLLaMA to better explore the potential of LLMs.

We demonstrate the efficacy and generalizability of our two-stage adaptation with extensive experiments on in-domain and out-domain datasets. We test our method on major decoder-only LLMs and various model scales, covering BLOOM 560M-7B (Scao et al., 2022), LLaMA-7B (Touvron et al., 2023b), Baichuan-7B (Yang et al., 2023), and Qwen-7B (Bai et al., 2023). The experimental results show that our method substantially improves over its baselines, highlighting the benefits of our progressive paradigm for text ranking. We also perform in-depth analysis into how our two-stage adaptation bridges the gap between LLMs and the text ranking task[2].

2 Method

2.1 Background

Text ranking refers to the task of determining how relevant each candidate document is to a given query. In our work, we exploit the pointwise strategy for inference, where the relevance scores are computed explicitly for each query-document pair (Crammer and Singer, 2001; Nogueira et al., 2019). The strategy has demonstrated high efficiency in real-world deployment compared with pairwise and listwise approaches (Liu et al., 2009).

Formally, given a query q and a set of candidate documents $D = \{d_1, \ldots, d_n\}$, we calculate relevance $\mathrm{score}(q, d_i), i \in [1, n]$ first and then execute a sorting procedure according to the scores. We can directly use the scoring methods of the out-of-the-box LLM exploration in text ranking to obtain $\mathrm{score}(q, d_i)$, which has shown significant effectiveness (Liang et al., 2022; Zhu et al., 2023).

Here we adopt one representative scoring strategy used in Sachan et al. (2022), treating the generation probability of q conditioned on d_i as the relevance score:

$$\mathcal{P}(d_i) = \text{`Document: } d_i \text{ Query:'}$$
$$\mathrm{score}(q, d_i) = \prod_j p\left(q_j \mid \mathcal{P}(d_i), q_{<j}\right). \quad (1)$$

Here, q_j denotes the j-th token of the query q, $q_{<j}$ represents the token sequence preceding the j-th token in query q, and $\mathcal{P}(d_i)$ represents the document-conditioned prompt. The calculation of each token generation probability can be parallelized, so the time complexity of this strategy is similar to that of its counterparts (Nogueira et al., 2019, 2020).

Although this out-of-the-box scoring method is mostly reasonable for text ranking, it would lead to suboptimal performance due to salient differences between the goals of LLM pre-training and text ranking, as illustrated in Figure 1. Previous studies have shown that we can better explore PLMs by adapting them with text ranking-specific training objectives (Nogueira et al., 2020; Zhuang et al., 2023b). Building on these studies, we propose a two-step training strategy, (1) CPT and (2) SFT, to adapt LLMs to text ranking, as shown in Figure 2.

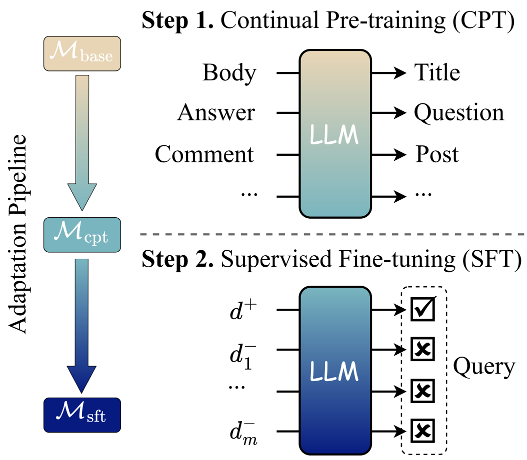

Figure 2: Two-stage adaptation paradigm. The base LLM $\mathcal{M}_{\text{base}}$ turns into an intermediate model $\mathcal{M}_{\text{cpt}}$ after CPT, and then $\mathcal{M}_{\text{cpt}}$ generates the final ranking model $\mathcal{M}_{\text{sft}}$ through SFT.

2.2 Continual Pre-training (CPT)

The first stage involves CPT of LLMs on a weakly supervised relevance dataset that is automatically collected. As originally proposed by Gururangan et al. (2020), CPT enables task- and domain-specific adaptation of LLMs for text ranking tasks.

A straightforward strategy of adapting LLMs to text ranking tasks is to perform SFT with ranking data, and RankLLaMA (Ma et al., 2023a) fol-

[2]The source code and models will be publicly available at https://github.com/xxx.

Text Pair Format	Source	Size	Query	Document
(title, body)	CommonCrawl	1.6M	Tango helps support North Texas Food Bank.	Tango Celebrates their 2013/2014 Partnership with the North Texas Food Bank …
(title, abstract)	arXiv	1.5M	Duality and Tameness.	We prove a duality theorem and show different kinds of failure of tameness of local cohomology.
(citation, reference)	Semantic Scholar	1.2M	Some comparative growth properties of composite entire and meromorphic functions …	The aim of this paper is to prove some results about composite entire and meromorphic functions …
(post, comment)	Reddit	1.7M	But are all evod 2 tanks glass?	The evod and evod 2 tanks are plastic. The evod glass …
(entity, description)	DBPedia	0.8M	Economy of Nigeria.	Nigeria is a middle income, mixed economy and …
(question, answer)	StackExchange	1.9M	How many chromosomes are in anaphase 2?	In anaphase II, the sister chromatids present at the end of meiosis I are separated into 23 individual chromosomes.
(summary, content)	CCNews	1.5M	Zidane apologizes for head butt.	French soccer star Zidane apologized for head-butting an Italian opponent …

Table 1: Examples of weakly supervised text pairs. Related words in queries and documents are highlighted in the same colors, showing that queries are often closely related to document content. Therefore, CPT on these data can help LLMs generate document-relevant queries, alleviating the misalignment shown in Figure 1.

lows this strategy. However, obtaining such large-scale and high-quality ranking datasets would be a formidable challenge. Moreover, large gaps between LLMs and ranking tasks may limit SFT's capability to fully explore the original knowledge in LLMs. A prospective solution is progressive multi-stage learning, where we first perform CPT to orient LLMs towards ranking goals before conducting final SFT for accurate alignment.

Weakly Supervised Data. Text relevance involves a range of aspects, such as question answering, semantic similarity, summarization, description. While LLM pre-training incorporate some of these aspects, here we further emphasize them since they are closely-related to text ranking.

To this end, we collect a large scale of text pairs covering different relevance types and domains as much as possible. Most text pairs are sourced from public web pages, mined through tailored protocols, and filtered via normalization. Concretely, we mine the following aspects of relevance to mock the query-document behaviors: (title, body), (title, abstract), (citation, reference), (post, comment), (entity, description), (question, answer) and (summary, content). Table 1 shows details and examples of the weakly supervised corpus.

Pre-training. We regard shorter texts, such as titles, posts, and summaries, as queries, and their corresponding longer texts as documents. Consistent with the typical pre-training goal of LLMs, we employ the NTP task on weakly supervised text pairs. The loss function $\mathcal{L}_{\text{ntp}}(q,d)$ is as follows:

$$\mathcal{L}_{\text{ntp}}(q,d) = -\sum_j \log p\left(q_j \mid \mathcal{P}(d), q_{<j}\right), \qquad (2)$$

which is equivalent to the log-likelihood of the relevance score defined in Eq.1.

2.3 Supervised Fine-tuning (SFT)

The second stage of our method is SFT, which helps further align the LLM for text ranking. SFT has been a common technique to accurately adapt LLMs for specific tasks (Ma et al., 2023a). Here we describe the supervised data first and then introduce the objectives for effective fine-tuning.

Supervised Training Data. We leverage the MS MARCO dataset for SFT, which comprises 8.8 million documents and 53,000 positive query-document pairs. Almost all positive text pairs in MS MARCO have been manually annotated, so this dataset is often used to train ranking models (Nogueira et al., 2019, 2020). In our work, we first employ the BGE model (Xiao et al., 2023) to retrieve the top 1000 negative document candidates (i.e., the most relevant documents) for each query. Following this, we construct the training dataset by randomly sampling corresponding positive and negative documents from the retrieved candidates.

The Ranking Objective. As demonstrated in previous text ranking studies (Ma et al., 2023a; Zhuang et al., 2023b), the ranking loss (Chen et al., 2020) based on a query q and the associated list of positive and negative documents $D = \{d^+, d_1^-, \ldots, d_m^-\}$ can effectively align LLMs with ranking tasks. The ranking loss is formulated as:

$$\mathcal{L}_{\text{rank}}(q,D) = -\log \frac{\exp(\mathcal{S}(q,d^+)/\tau)}{\sum_{d\in D}\exp(\mathcal{S}(q,d)/\tau)}, \qquad (3)$$

where τ denotes the temperature parameter, and $\mathcal{S}(.)$ is the relevance scoring function.

Our objective differs from that of Ma et al. (2023a) as we use score(.) in Eq. 1 as the scoring function $\mathcal{S}(.)$ rather than exploiting the last-token representation for relevance scoring. score(.) incorporates more scoring evidence by considering all query tokens. More importantly, our objective is to directly optimize NTP probabilities, a key property of LLMs, to fit the ranking goal. In this way, we can benefit from the strong generalization capabilities of pre-trained LLMs.

Auxiliary Objectives. LLMs have potential for better handling out-domain scenarios as they are pre-trained on a large and diverse corpora (OpenAI, 2023). Yet the large-scale parameters of LLMs might cause overfitting to the training dataset. To avoid this problem, we supplement the ranking optimization with two additional objectives. The first one is the NTP objective $\mathcal{L}_{\text{ntp}}$, which is consistent with CPT and utilizes positive text pairs in SFT data. The second one is newly designed by us, namely Differential Penalty (DP) as follows:

$$\mathcal{L}_{\text{dp}}(\mathcal{M}_{\text{cpt}}, \mathcal{M}_{\text{sft}}) = \frac{1}{\|T\|} \sum_{j}^{\|T\|} \sum_{k}^{\|V\|} \text{KL}(p_{\text{cpt}}^{j,k}, p_{\text{sft}}^{j,k}), \quad (4)$$

where KL is the Kullback-Leibler (KL) divergence, V is the model vocabulary, and T is all query tokens. $p_{\text{cpt}}^{j,k}$ and $p_{\text{sft}}^{j,k}$ denote the token probabilities calculated by the model $\mathcal{M}_{\text{cpt}}$ and the model $\mathcal{M}_{\text{sft}}$ respectively. The DP objective actually constrains the generation difference between the adapted model and the initialized model.

Overall, our mixed objective loss function during SFT is as follows:

$$\mathcal{L}_{\text{sft}} = \alpha \mathcal{L}_{\text{rank}} + (1 - \alpha)(\mathcal{L}_{\text{ntp}} + \mathcal{L}_{\text{dp}}), \quad (5)$$

where α is the trade-off hyper-parameter.

3 Experiments

3.1 Experimental Settings

Test Datasets. We use the same experimental settings as Ma et al. (2023a), covering both in-domain and out-domain scenarios.

For the in-domain scenario, we test on MS MARCO (Nguyen et al., 2016), DL19 (Craswell et al., 2020) and DL20 (Craswell et al., 2021) benchmarks, and construct candidate documents based on the top 1000 documents retrieved by BM25 (Robertson and Zaragoza, 2009) and the top 200 documents retrieved by BGE (Xiao et al., 2023) respectively.

For the out-domain scenario, we test on BEIR benchmark (Thakur et al., 2021) and use the top 1000 documents retrieved by BM25 as candidate documents. The BEIR benchmark covers a variety of domains and ranking tasks, and therefore could be used to measure the generalization ability of ranking models.

Implementation Details. We train the model on 8 NVIDIA A100 GPUs with 80GB of memory. During CPT, we train for 1 epoch on all weakly supervised data. During SFT, we train for 1 epoch on the MS MARCO training set. Following Eq. 3, we set the number of negative examples m to 48 and the temperature parameter τ to 0.001. The trade-off hyper-parameter α in Eq. 5 is set to 0.6. Similar to previous work (Ma et al., 2023a), we fine-tune the top 16 transformer layers and freeze other parameters to reduce GPU memory of SFT.

Baselines and Metric. We compare text ranking models with different structures in previous works, including encoder-only MonoBERT (Nogueira et al., 2019), encoder-decoder MonoT5 (Nogueira et al., 2020) and RankT5 (Zhuang et al., 2023b), and decoder-only RankLLaMA (Ma et al., 2023a). Following standard practice, we adopt NDCG@10 as the evaluation metric.

3.2 Main Results

To verify the broad effectiveness of our method, we compare its performance with other adaptation approaches on four foundation LLMs of different types and sizes: BLOOM 560M-7B (Scao et al., 2022), LLaMA-7B (Touvron et al., 2023b), Qwen-7B (Bai et al., 2023) and Baichuan-7B (Yang et al., 2023). Our **t**wo-**s**tage **a**daptation of **LLM**s for text **rank**ing is denoted as TSARankLLM.

In-Domain Evaluation. Table 2 summarizes the in-domain performance of our models as well as several previous works. We find that decoder-only LLMs exhibit substantial ranking capabilities. The proposed TSARankLLM models outperform MonoT5 models of similar scale in almost all cases. Aligning with prior works (Nogueira et al., 2020), we observe ranking performance generally increase with model size. For example, increasing the size of BLOOM from 560M to 7B improves the average NDCG@10 score by $66.3 - 64.3 = 2.0$.

Following, we look into our TSARankLLM models in terms of different foundation LLMs. By comparing these models of the same 7B-scale parameter size, we can see that LLaMA-7B achieves the best performance overall, surpassing BLOOM, Baichuan and Qwen. While other foundation LLMs occasionally surpass LLaMA-7B on certain datasets (e.g. BLOOM-7B on DL19 and Qwen-7B on MS MARCO), LLaMA-7B appears to be the most effective foundation LLM of those examined.

Method	LLM	Size	Sparse Retrieval - BM25			Dense Retrieval - BGE			Average
			MS MARCO	DL19	DL20	MS MARCO	DL19	DL20	
Retrieval	NA	NA	22.8	50.6	48.0	40.9	71.4	70.5	50.7
MonoBERT	BERT	340M	44.0	72.3	70.3	44.7	72.0	70.2	62.3
MonoT5	T5	220M	43.6	71.5	69.7	43.4	69.4	65.8	60.6
	T5	770M	43.4	73.2	71.2	43.5	72.0	70.1	62.2
	T5	3B	44.9	72.8	74.5	45.7	72.5	74.5	64.2
RankLLaMA	LLaMA	7B	46.9	74.4	**76.4**	47.9	74.7	76.2	66.1
TSARankLLM	BLOOM	560M	44.0	75.3	73.2	44.8	75.0	73.7	64.3
	BLOOM	1B	44.5	75.6	72.3	45.4	75.4	72.9	64.4
	BLOOM	3B	45.1	76.8	73.6	45.9	76.2	74.4	65.3
	BLOOM	7B	46.0	**77.3**	74.6	47.0	**77.1**	75.9	66.3
	LLaMA	7B	46.6	76.2	76.3	47.7	76.7	**76.8**	**66.7**
	Baichuan	7B	46.6	75.9	74.3	47.7	75.2	76.2	66.0
	Qwen	7B	**48.0**	75.8	74.3	**49.0**	75.5	75.0	66.3

Table 2: In-domain results of various models.

Dataset	BM25	MonoBERT	MonoT5			RankT5	TSARankLLM		
	NA	BERT	T5			T5	BLOOM		
	NA	340M	220M	770M	3B	770M	560M	1B	3B
Arguana	39.7	51.5	13.2	30.2	28.8	33.0	53.3	55.1	**55.6**
Climate	16.5	24.9	24.5	25.9	**28.0**	21.5	22.3	23.6	27.7
DBPedia	31.8	43.5	42.0	43.5	47.8	44.2	44.5	45.4	**50.0**
FEVER	65.1	81.3	80.2	82.8	**85.0**	83.2	83.6	82.3	83.7
FiQA	23.6	36.8	41.4	44.6	**51.4**	44.5	40.0	42.0	44.9
HotpotQA	63.3	73.5	69.5	73.6	75.9	71.0	75.6	75.4	**76.4**
NFCorpus	33.8	36.9	35.7	38.4	38.4	38.1	37.3	37.9	**39.4**
NQ	30.6	56.8	56.7	60.8	**63.3**	61.4	56.1	57.6	58.7
Quora	78.9	71.5	82.3	**85.4**	84.1	83.1	82.9	82.3	82.9
SCIDOCS	14.9	15.7	16.5	19.1	**19.7**	18.1	18.1	18.7	19.2
SciFact	67.9	72.0	73.6	75.5	77.7	75.0	77.1	76.3	**78.0**
COVID	59.5	65.0	77.8	82.3	79.5	80.7	78.9	81.7	**83.3**
Touche	**44.2**	27.7	27.7	28.5	30.0	44.0	28.2	29.7	30.2
#Best	1	0	0	1	5	0	0	0	**6**
Average	48.3	50.5	49.3	53.1	54.6	53.7	53.7	54.5	**56.2**

Table 3: Out-domain results of 220M-3B models. The "# Best" row indicates the number of datasets on which each model achieved the best performance.

Finally, we compare TSARankLLM and RankLLaMA adaptations on the same LLaMA-7B foundation. As shown, our two-stage adaptation approach surpasses the single-stage SFT of RankLLaMA, yielding average improvements of $66.7 - 66.1 = 0.6$ NDCG@10. Overall, our TSARankLLM method can provide state-of-the-art performance for the in-domain setting.

Out-Domain Evaluation. Table 3 shows the results of models within 3B scale, while Table 4 provides the results of 7B models. In general, out-domain performance significantly lags behind that of the in-domain, with a gap around 10 points, indicating that out-domain text ranking is challenging.

After examining the results in Table 3, we find that the overall tendency is consistent with that of in-domain results. Increased model size generally improves performance. Notably, our TSARankLLM model based on BLOOM-3B outperforms the monoT5-3B of the same model size, demonstrating its strong text ranking capability.

Further, we analyze the out-domain results of 7B-scale models in Table 4. We can see that our TSARankLLM models exhibit much better performance than RankLLaMA. With the same LLaMA-7B as backend, our method gets higher NDCG@10 scores on all the datasets, and the averaged increase reaches $57.8 - 52.5 = 5.3$. Among the foundation models, Qwen achieves the highest average score with our TSARankLLM approach, while LLaMA excels on most out-domain datasets. Overall, all the results indicate that LLMs provide a promising solution for out-domain text ranking if pre-trained LLM knowledge can be effectively explored.

Dataset	RankLLa.	TSARankLLM			
	LLa.	BLO.	LLa.	Bai.	Qwen
Arguana	47.0	56.1	51.2	54.1	**56.8**
Climate	19.1	28.1	31.6	32.2	**37.2**
DBPedia	48.6	47.8	49.2	**50.0**	48.5
FEVER	74.5	84.1	84.5	83.0	**85.6**
FiQA	42.2	46.4	**50.2**	49.4	49.6
HotpotQA	75.2	77.4	**80.7**	80.0	80.4
NFCorpus	35.8	39.5	40.2	39.8	**40.4**
NQ	62.1	60.4	**63.6**	63.1	62.7
Quora	80.5	83.8	**85.9**	85.0	82.6
SCIDOCS	19.0	**20.3**	19.8	20.0	19.8
SciFact	70.1	77.8	**78.3**	77.6	77.6
COVID	77.4	82.9	**84.0**	82.4	83.7
Touche	31.0	31.0	32.0	30.6	**32.6**
#Best	0	1	**6**	1	5
Average	52.5	56.6	57.8	57.5	**58.3**

Table 4: Out-domain results of 7B models. "RankLLa.", "BLO.", "LLa." and "Bai." represent RankLLaMA, BLOOM, LLaMA and Baichuan respectively.

Method	BLOOM-560M		LLaMA-7B	
	In.	Out.	In.	Out.
TSARankLLM	**64.3**	**53.7**	**66.7**	**57.8**
• *Two-Stage Training*				
- CPT	63.8	51.0	66.3	55.7
- SFT	53.6	47.8	56.3	52.6
- CPT&SFT	43.4	43.4	47.5	48.6
• *Auxiliary Objectives of SFT*				
- $\mathcal{L}_{\text{ntp}}$	**64.3**	53.2	66.6	57.0
- $\mathcal{L}_{\text{dp}}$	64.2	52.7	66.4	56.3
- $\mathcal{L}_{\text{ntp}}$&$\mathcal{L}_{\text{dp}}$	64.2	52.0	66.4	55.5

Table 5: Ablation results of BLOOM-560M and LLaMA-7B in in-domain (In.) and out-domain (Out.) scenarios.

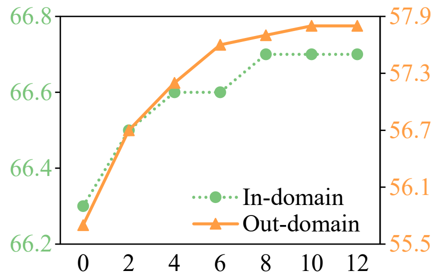

Figure 3: Results of TSARankLLM based on LLaMA-7B at various CPT data sizes (millions).

3.3 Ablation Analysis

In this subsection, our ablation analysis quantifies the contribution of each part of our two-stage adaptation to the overall performance improvements of the TSARankLLM model.

The Two-Stage Training. Table 5 shows the individual contributions of CPT and SFT based on BLOOM-560M and LLaMA-7B. As shown, both training stages exhibit significant performance gains, validating their importance to optimal final results. In particular, the influence of SFT is highly remarkable. This is unsurprising given its use of high-quality training data from MS MARCO to accurately align LLMs with ranking objectives. Without both stages, our models degenerate to out-of-the-box LLMs with unsatisfactory ranking capabilities, whose in-domain performance can be even worse than direct retrieval without ranking. This directly indicates that the misalignment between out-of-the-box LLMs and text ranking objectives results in suboptimal performance.

The Scale of Weakly-Supervised Data in CPT. During CPT, we construct a weakly-supervised dataset. An important question is that how the data scale influences our model performance. Figure 3 shows the results of our method on LLaMA-7B, illustrating trends for both in-domain and out-domain performance. As expected, model performance improves with greater data scale, though the gains become insignificant after the data scale surpasses 10M. Additionally, we can see that CPT can benefit the out-domain performance more than the in-domain setting, as evidenced by the steeper curve of the out-domain case.

The Ranking Objective of SFT. As mentioned in Section 2.3, we exploit a ranking objective different from that of RankLLaMA. Here we fairly compare both ranking objectives in two settings: with and without CPT. Both settings during SFT only exploit the ranking objective, i.e., the auxiliary objectives are removed. Figure 4 shows that our ranking objective substantially outperforms that of RankLLaMA in both settings in terms of either in-domain or out-domain performance. The results indicate that our SFT can better align LLM pre-training with the ranking goal by optimizing the full-query generation probabilities directly.

The Auxiliary Objectives of SFT. To avoid overfitting and better explore the potential of LLMs, we design two auxiliary objectives (i.e., $\mathcal{L}_{\text{ntp}}$ and $\mathcal{L}_{\text{dp}}$) during SFT. Table 5 conducts ablation analysis to test the effectiveness of the two objectives based on BLOOM-560M and LLaMA-7B. We can see that the two objectives have negligible impact on in-domain performance, with a maximum drop of only 0.3 point when removed. As expected, they signif-

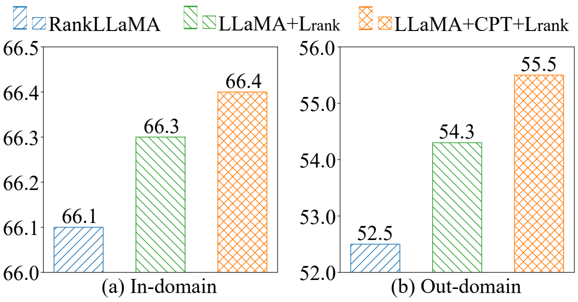

Figure 4: Model results with different ranking objectives. RankLLaMA is based on the last token, while two variants of our models are based on entire query tokens and do not involve our two auxiliary objectives.

Dataset	Pairwise	Listwise		TSARankLLM
	UL2-20B	ChatGPT	GPT-4	LLaMA-7B
COVID	79.5	76.7	**85.5**	84.0
NFCorpus	36.1	35.6	38.5	**40.2**
Touche	37.9	36.2	**38.6**	32.0
DBPedia	46.5	44.5	47.1	**49.2**
SciFact	73.3	70.4	75.0	**78.3**
#Best	0	0	2	**3**
Average	54.7	52.7	**56.9**	56.7

Table 6: Comparison with more powerful LLMs. Due to high time complexity and API call costs of pairwise and listwise ranking strategies (Qin et al., 2023), we only experiment with small datasets.

Method	Doc-word ↑	Stop-word ↓
Base LLaMA-7B	25.1	19.7
+ CPT	27.8	16.1
+ SFT	28.3	15.7
+ CPT&SFT	31.7	14.1
GPT-4	**33.6**	13.6
Ground-Truth	33.3	**13.3**

Table 7: The proportion of document-relevant and stop words in queries generated by various models. "Ground-Truth" denotes manually annotated positive queries.

icantly impact the out-domain performance, decreasing performance by $\frac{(53.7-52.0)+(57.8-55.5)}{2} = 2.0$ points on average when omitted. Notably, removing $\mathcal{L}_{\text{dp}}$ leads to a more substantial decrease, indicating its greater impact. Moreover, we find that LLaMA-7B is more sensitive to the auxiliary objectives than BLOOM-560M, probably because larger models can be more easily overfitted.

3.4 Discussion

In this subsection, we conduct detailed experimental analyses to comprehensively evaluate our two-stage training method.

A Comparison with More Powerful LLMs. Compared to the LLMs investigated in our work, some recent LLMs like UL2-20B (Tay et al., 2023), ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023) have shown more superior performance on many NLP tasks owing to their extremely large model sizes and pre-training data. We apply our TSARankLLM method to train the LLaMA-7B, and compare its performance against these LLMs. Due to the massive model sizes and even closed-source nature of these LLMs, directly training them is challenging. Here we apply out-of-the-box ranking strategies to them: (1) the pairwise ranking strategy of Qin et al. (2023) based on UL2-20B and (2) the listwise ranking strategy of Sun et al. (2023b) based on ChatGPT and GPT4. Table 6 shows the results on five out-domain datasets selected for this comparison[3]. We can see that our method is very competitive, with an average gap of only $56.9 - 56.7 = 0.2$ point compared to GPT-4. Considering the higher costs of these compared systems, our method could be preferable in practice.

[3]The in-domain evaluation is unfair for out-of-the-box ranking strategies.

Quality Assessment of Query-Generation in Our Two-Stage Training. Query generation is a key module in our two-stage training. To understand this module, we conduct a human evaluation to measure the quality of query-generation directly. As mentioned in Figure 1, without text-ranking-oriented alignment, the out-of-the-box LLMs commonly generate queries including a large proportion of document-irrelevant information. As such, we mainly measure the quality by the percentage of document-relevant semantically-equivalent words, while also considering adverse stop words. Table 7 shows the manual evaluation results on 500 randomly selected BEIR samples. The results of GPT4 and ground-truth are also provided for reference. We can see that by applying CPT and SFT gradually, the percentage of document-relevant words increases while that of stop words decreases, both approaching the percentages of GPT4 and ground-truth. However, higher percentage of document-relevant words and lower percentage of stop words do not necessarily indicate better performance.

A Perplexity Perspective to Examine Our Two-Stage Training. An optimal ranker based on Eq. 1 should strongly prefer generating positive over negative queries for a given document. Therefore, the perplexity (denoted as PPL) difference

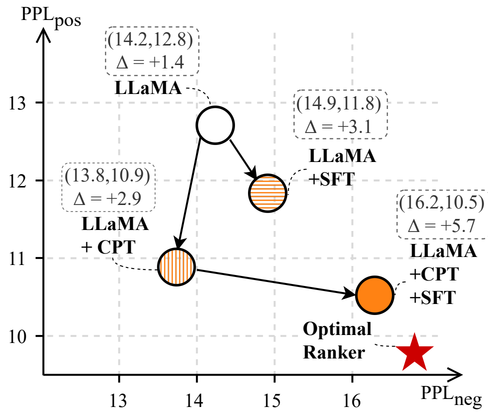

Figure 5: The perplexity (PPL) comparison of generating positive queries for a given document. "Δ" denotes $\mathrm{PPL}_{\mathrm{neg}} - \mathrm{PPL}_{\mathrm{pos}}$, which roughly indicates the model's ranking capability.

between negative and positive queries can roughly estimate the ranking capability of our models. We randomly select 5,000 positive and negative text pairs from BEIR and test the query-generation PPL of various rankers, as shown in Figure 5. We observe that in our two-stage training pipeline, CPT significantly reduces the PPL of positive queries (i.e, $\mathrm{PPL}_{\mathrm{pos}}$), and SFT increases PPL of negative queries (i.e., $\mathrm{PPL}_{\mathrm{neg}}$) remarkably. The observation is consistent with our initial expectation.

4 Related Work

Text ranking has been an active area of research for decades (Liu et al., 2009; Zhu et al., 2023). One widely-adopted strategy is to rank candidate documents based on the relevance score between a query and the document (Cossock and Zhang, 2006). This strategy is referred to as the pointwise approach. Additionally, pairwise (Burges et al., 2005) and listwise (Xia et al., 2008) approaches, which consider the relative ordering of multiple documents in response to a query, have also gained great attention due to their strong performance. In our work, we utilize the pointwise approach for efficient inference and the listwise approach for effective training, making the best use of the both.

The precise calculation of the query-document relevance scores is the key to the pointwise inference, which has been generally dominated by supervised techniques (Guo et al., 2016). Initially, manually-crafted sparse features were employed to estimate these scores (Chapelle and Chang, 2011). Subsequently, neural network models marked a turning point, showcasing their substantial promise (Pang et al., 2016). More recently, the progress of PLMs has led to remarkable advances in score calculation by using a pre-training and fine-tuning framework for text ranking (Gao et al., 2021; Ju et al., 2021; Pradeep et al., 2021; Zhang et al., 2023b; Li et al., 2023).

As PLMs evolved into decoder-only LLMs, early work explored out-of-the-box strategies for text ranking to leverage the inherent strong reasoning capabilities of LLMs (Sun et al., 2023b; Qin et al., 2023; Ma et al., 2023b; Cho et al., 2023). For instance, one could simply ask LLMs to determine the relevance of a query-document pair, yielding a rudimentary solution (Liang et al., 2022; Zhuang et al., 2023a,c). Subsequent works (Sachan et al., 2022; Muennighoff, 2022; Drozdov et al., 2023), propose the use of query generation likelihood based on a candidate document as a measure of relevance. We follow this line of work for the relevance score definition.

Nevertheless, these out-of-the-box strategies often overlook the potential misalignment between LLMs and the specific requirements of text ranking tasks, which is a major issue that our work aims to address. Task-specific LLM training can help bridge this gap (Sun et al., 2023a; Ma et al., 2023a; Zhang et al., 2023a), as exemplified by RankLLaMA (Ma et al., 2023a), which uses the last token as the ranking basis and trains with ranking losses.

While dominant LLMs typically feature a decoder-only architecture, previous research on adapting encoder(-decoder) PLMs to text ranking remains highly pertinent (Nogueira et al., 2019, 2020; Zhuang et al., 2023b). The underlying principles and core ideas of these studies underpin our approach, and their benefits apply across diverse model architectures.

5 Conclusion

In our work, we proposed a novel two-stage training paradigm to adapt LLMs to text ranking tasks. Specifically, we first performed CPT on a large-scale weakly-supervised corpus to initially align LLMs with ranking objectives. This is then followed by SFT on high-quality data along with full-query generation optimization and auxiliary objectives. Through this two-stage training paradigm, we achieved improved ranking performance on various LLMs in both in-domain and out-domain experimental settings. The significant gains exhibited by our approach highlight its effectiveness in enhancing the capabilities of LLMs for text ranking.

6 Limitations

While our two-stage adaptation can effectively enhance LLMs' text ranking capabilities, several limitations remain. First, our CPT is actually independent of ranking objectives. Introducing ranking objectives similar to SFT during CPT worths further exploration. Second, using a unified prompt for all ranking tasks might damage model generalization. We generally refer to the texts of the BEIR benchmark as "query" and "document" in prompts, as shown in Eq. 1. In fact, these texts can be further classified. Specifically, "query" involves "title", "entity", etc, and "document" involves "argument", "news", etc. Finally, we evaluate our approach only on the BEIR benchmark. Testing it on diverse ranking tasks, such as the demonstration ranking of the in-context-learning scenario and knowledge ranking to mitigate hallucinations in LLMs, would better validate its broad applicability.

7 Ethics Statement

Text ranking is a widely studied task. The data and other related resources involved in our work are open source and are already commonly used in existing work. To the best of our knowledge, our work fully complies with the ACL Ethics Policy.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Olivier Chapelle and Yi Chang. 2011. Yahoo! learning to rank challenge overview. In *Proceedings of the Yahoo! Learning to Rank Challenge, held at ICML 2010, Haifa, Israel, June 25, 2010*, volume 14 of *JMLR Proceedings*, pages 1–24. JMLR.org.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Sukmin Cho, Soyeong Jeong, Jeongyeon Seo, and Jong C. Park. 2023. Discrete prompt optimization via constrained generation for zero-shot re-ranker. In *ACL (Findings)*, pages 960–971. Association for Computational Linguistics.

David Cossock and Tong Zhang. 2006. Subset ranking using regression. In *Learning Theory: 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006. Proceedings 19*, pages 605–619. Springer.

Koby Crammer and Yoram Singer. 2001. Pranking with ranking. *Advances in neural information processing systems*, 14.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR*, abs/2102.07662.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR*, abs/2003.07820.

Andrew Drozdov, Honglei Zhuang, Zhuyun Dai, Zhen Qin, Razieh Rahimi, Xuanhui Wang, Dana Alon, Mohit Iyyer, Andrew McCallum, Donald Metzler, et al. 2023. Parade: Passage ranking using demonstrations with large language models. *arXiv preprint arXiv:2310.14408*.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Proc. of ECIR*.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*, pages 55–64. ACM.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*, pages 8342–8360. Association for Computational Linguistics.

Jia-Huei Ju, Jheng-Hong Yang, and Chuan-Ju Wang. 2021. Text-to-text multi-view learning for passage re-ranking. In *SIGIR*, pages 1803–1807. ACM.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Zhen Li, Chongyang Tao, Jiazhan Feng, Tao Shen, Dongyan Zhao, Xiubo Geng, and Daxin Jiang. 2023. Faa: Fine-grained attention alignment for cascade document ranking. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1688–1700.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel J. Orr, Lucia Zheng, Mert Yüksekgönül, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models. *CoRR*, abs/2211.09110.

Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Simon Lupart, Thibault Formal, and Stéphane Clinchant. 2023. Ms-shift: An analysis of ms marco distribution shifts on neural retrieval.

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

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

Niklas Muennighoff. 2022. SGPT: GPT sentence embeddings for semantic search. *CoRR*.

Tri Nguyen, Mir Rosenberg, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proc. of NeurIPS*.

Rodrigo Frassetto 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, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 708–718. Association for Computational Linguistics.

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. Gpt-4 technical report.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. *CoRR*, abs/1602.06359.

Ronak Pradeep, Rodrigo Frassetto Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *CoRR*, abs/2101.05667.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Devendra Singh 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, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3781–3797. Association for Computational Linguistics.

Teven Le Scao et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Weiwei Sun, Zheng Chen, Xinyu Ma, Lingyong Yan, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023a. Instruction distillation makes large language models efficient zero-shot rankers.

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

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Siamak Shakeri, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. Ul2: Unifying language learning paradigms.

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 *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199.

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

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models.

Xinyu Zhang, Sebastian Hofstätter, Patrick Lewis, Raphael Tang, and Jimmy Lin. 2023a. Rank-without-gpt: Building gpt-independent listwise rerankers on open-source large language models. *CoRR*, abs/2312.02969.

Zongmeng Zhang, Wengang Zhou, Jiaxin Shi, and Houqiang Li. 2023b. Hybrid and collaborative passage reranking. *arXiv preprint arXiv:2305.09313*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2023a. Beyond yes and no: Improving zero-shot LLM rankers via scoring fine-grained relevance labels. *CoRR*, abs/2310.14122.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023b. Rankt5: Fine-tuning T5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2308–2313. ACM.

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

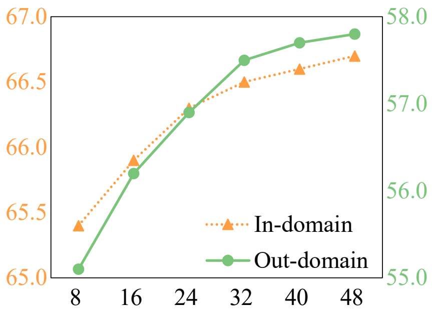

Figure 6: Results for various negative examples sizes m in Eq. 5 on LLaMA-7B.

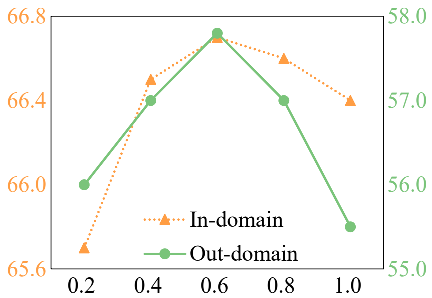

Figure 7: The effect of α in Eq. 5 on LLaMA-7B.

A Appendix

The Negative Examples Size. Negative example size m in Eq. 5 is the key to ranking loss. We test the impact of m on in-domain and out-domain performance in Figure 6. Overall, increasing negative examples enhances ranking performance, consistent with traditional conclusions (Zhuang et al., 2023b; Ma et al., 2023a). Moreover, we find that out-domain performance is more sensitive to negative example size. Specifically, as the m increases from 8 to 48, the in-domain result increases by 1.3, while the corresponding increase in the out-domain result is 2.7.

Balancing Ranking and Auxiliaries. During SFT, the trade-off between ranking and auxiliary objectives is controlled by α in Eq. 5. We analyze the impact of α, as shown in Figure 7. If the α is too low, the model would ignore the ranking objective, which is the key to SFT. Therefore, when the α is 0.2, the model performs poorly in both scenarios. Conversely, high α values barely affect in-domain results. From α=0.6 to 1.0, the in-domain performance only decreases by 0.3. However the corresponding out-domain result drops significantly by 2.3. This is because over-emphasis on the ranking objective leads to overfitting in the in-domain scenario and affects model generalization.

Method	In-domain		Out-domain	
	Short	Long	Short	Long
MonoT5-3B	65.7	61.2	57.4	52.4
RankLLaMA-7B	67.3	63.5	55.1	50.4
• *TSARankLLM*				
BLOOM-3B	66.9	64.2	57.5	55.2
LLaMA-7B	**67.6**	**66.0**	**58.6**	**57.2**

Table 8: A comparison of models on query sets of short ($\leq$10 words) and long ($>$10 words) lengths.

A fine-grained comparison of text ranking by query-length. Query length is known to greatly impact the ranking performance (Lupart et al., 2023). Table 8 provides a comparison of two TSARankLLM models against two representative baselines across varying query lengths. We find that our TSARankLLM models excel particularly on long queries, with most overall gains attributable to the improvements on long queries. Notably, our method exhibits little performance degradation across query lengths under the out-domain setting.

Efficiency Analysis We analyze the number of text pairs processed per second for various models, as shown in Table 9. All experiments are performed on 1 NVIDIA A100 GPU with 80GB of memory. We find that model size and batch size to be the primary determinants of inference efficiency, rather than the model structure or scoring strategy investigated. Although our scoring strategy is different from other models, MonoBERT-340M, MonoT5-770M and TSARankLLM BLOOM-560M achieve comparable inference efficiency at similar model scales. The same is true for MonoT5-3B and TSARankLLM BLOOM-3B. Unexpectedly, a larger batch size can significantly improve the inference efficiency.

Batch	MonoBERT	MonoT5			TSARankLLM			
	BERT	T5			BLOOM			
	340M	220M	770M	3B	560M	1B	3B	7B
1	56	57	51	30	53	48	32	21
2	108	112	95	54	100	87	56	35
4	172	175	157	86	168	139	86	53
8	263	270	231	114	256	197	116	64

Table 9: Number of text pairs processed per second by encoder-only (MonoBERT), encoder-decoder(MonoT5), decoder-only (TSARankLLM) models.