Leveraging Reference Documents for Zero-Shot Ranking via Large Language Models

Jieran Li[1], Xiuyuan Hu[1], Yang Zhao[1], Shengyao Zhuang[2], Hao Zhang[1]

[1]Department of Electronic Engineering, Tsinghua University

[2]CSIRO

{lijr23,huxy22}@mails.tsinghua.edu.cn

{zhao-yang,haozhang}@tsinghua.edu.cn

shengyao.zhuang@csiro.au

Abstract

Large Language Models (LLMs) have demonstrated exceptional performance in the task of text ranking for information retrieval. While Pointwise ranking approaches offer computational efficiency by scoring documents independently, they often yield biased relevance estimates due to the lack of inter-document comparisons. In contrast, Pairwise methods improve ranking accuracy by explicitly comparing document pairs, but suffer from substantial computational overhead with quadratic complexity ($O(n^2)$). To address this tradeoff, we propose **RefRank**, a simple and effective comparative ranking method based on a fixed reference document. Instead of comparing all document pairs, RefRank prompts the LLM to evaluate each candidate relative to a shared reference anchor. By selecting the reference anchor that encapsulates the core query intent, RefRank implicitly captures relevance cues, enabling indirect comparison between documents via this common anchor. This reduces computational cost to linear time ($O(n)$) while importantly, preserving the advantages of comparative evaluation. To further enhance robustness, we aggregate multiple RefRank outputs using a weighted averaging scheme across different reference choices. Experiments on several benchmark datasets and with various LLMs show that RefRank significantly outperforms Pointwise baselines and could achieve performance at least on par with Pairwise approaches with a significantly lower computational cost.

1 Introduction

Large language models (LLMs), such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2023), Llama 2 (Touvron et al., 2023), and FlanT5 (Wei et al., 2021), have demonstrated remarkable performance across a range of natural language processing tasks, particularly in zero-shot prompting settings. Notably, these LLMs have been specifically designed with prompts for zero-shot docu-
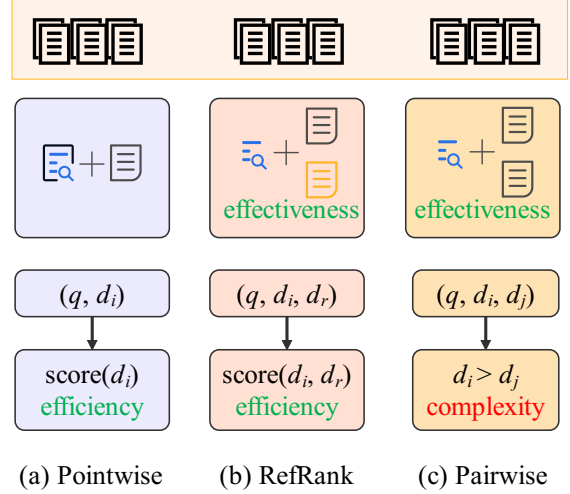

Figure 1: Comparison of three sorting methods: (a) Pointwise, (b) RefRank, (c) Pairwise.

ment ranking (Agrawal et al., 2023; Kojima et al., 2022; Wang et al., 2023). By utilizing diverse prompt designs, these LLMs generate relevance estimates for each candidate document, thereby enabling effective document ranking. Unlike traditional neural ranking methods (Yates et al., 2021), LLM-based rankers do not require additional supervised fine-tuning and exhibit robust capabilities in zero-shot ranking scenarios.

Pointwise (Liang et al., 2022; Zhuang et al., 2023a,b; Sachan et al., 2022; Guo et al., 2025) and Pairwise (Qin et al., 2023) are two fundamental strategies in the zero-shot ranking for LLMs. Pointwise methods predict the relevance probability between query q and document d, ranking documents based on their log-likelihood values (Rivera-Garrido et al., 2022; Zhuang et al., 2023a). Although the Pointwise methods have a high efficiency, their ranking performance is often limited due to the lack of direct comparisons between different documents. In contrast, Pairwise methods compare the relevance of documents in pairs relative to a query, aggregating the results to achieve

1

a ranking (Qin et al., 2023). While these methods can produce more effective ranking results, their typical time complexity of $O(n^2)$ makes them impractical for real-world scenarios (Rivera-Garrido et al., 2022; Zhuang et al., 2023a).

Inspired by the control groups used in scientific experiments (James, 1980), we propose **RefRank**, a reference-based ranking paradigm for documents. Figure 1 illustrates the core idea of RefRank and compares it to the Pointwise and Pairwise approaches. RefRank selects one document from the initially sorted candidate documents to serve as a reference document r, extending the query-document pair (q, d) into a triplet (q, d, r). This design enables the LLM to score documents based on the reference document, establishing indirect comparative relationships among documents, while maintaining the same time complexity as the Pointwise approaches.

The quality of rankings among various reference documents is inconsistent. Inspired by the concept of pseudo-relevance feedback (PRF) in information retrieval, which assumes that the top-ranked documents from any retrieval phase contain relevant signals for query rewriting (Xu and Croft, 2017), we utilize the top-ranked documents as reference documents for our RefRank. Using experimental methods, we performed a statistical analysis across four datasets: TREC-DL2019 (Craswell et al., 2020), TREC-DL2020 (Craswell et al., 2021), Signal and News (Thakur et al., 2021). Our findings indicate that selecting documents retrieved in the top ranks during the initial phase of the retrieval process typically results in higher ranking quality. In practical applications, it is advisable to prioritize the selection of the top two documents as a reference document. Building on this foundation, we propose the integration of higher-ranking results through evaluation weighting to further enhance ranking quality. Ultimately, we implement two variants of RefRank: a single-ranking strategy and a multi-weighted strategy.

We conducted a comparative analysis of existing principal methodologies using FLAN-T5 as the backbone model across the TREC-DL 2019, TREC-DL 2020, and BEIR datasets. Our proposed method achieved optimal performance on several test sets within these datasets. Additionally, we further validated the efficacy of our approach by employing decode-only models, specifically Llama 3.1 and Qwen 2.5, on the TREC-DL 2019 and TREC-DL 2020 datasets.

We note that the "Pointwise" and "Pairwise" paradigms are two common and classic approaches in ranking tasks. The novelty of our work lies in the clever integration of the advantages of both paradigms by selecting a reference document for ranking. The contributions of this paper can be summarized as follows:

1. We propose **RefRank**, a new reference-based paradigm for document ranking. By comparing and evaluating candidate documents against reference documents, we achieve a balance of efficiency and effectiveness.

2. We utilize ranking information retrieved from the first stage to select reference documents.

3. We present a method for the average weighted integration of high-quality evaluation results, which further enhances ranking quality and explores the potential for weighting similar methods.

4. Through systematic and comprehensive experimental analyses, we validate the effectiveness of the proposed method.

2 Related Works

2.1 Pointwise Approaches

Pointwise approaches assess and rank documents by quantifying the relevance of a query q to each document. Current mainstream methods can be divided into two categories: (1) direct relevance assessment (Liang et al., 2022; Zhuang et al., 2023a); (2) query similarity assessment generated (Zhuang et al., 2023b). The first method employs query-document pairs as input to the model, prompting the LLM to produce a binary label ("yes"/"no"). The normalized log-likelihood value assigned to "yes" is subsequently used as the document score. To improve evaluation accuracy, this method can be extended to accommodate multiple labels, such as three or four labels (Zhuang et al., 2023a). The second method commences by generating candidate queries from the document utilizing a large language model (LLM). Following this, it computes the semantic similarity between the generated queries and the original query. This procedure entails two inference operations from the LLM: the first generates the queries, while the second evaluates their similarity to the original query. Pointwise approaches generate rankings by independently

evaluating individual documents. The primary advantage of this method is its ability to utilize output logits produced by LLMs for precise scoring. However, the lack of relative comparisons among documents considerably reduces overall effectiveness.

2.2 Pairwise Approaches

Pairwise approaches take a query q and a pair of documents (d_i, d_j) as input. These approaches employ prompt learning with an LLM to assess the relevance of the two documents, ultimately determining and outputting the more relevant document—either d_i or d_j (Qin et al., 2023). Theoretically, this method has the potential to achieve higher ranking accuracy by performing Pairwise comparisons across all candidate documents. However, its time complexity is $O(n^2)$, which renders it computationally demanding. To enhance efficiency, existing research has proposed various sampling strategies (Gienapp et al., 2022; Mikhailiuk et al., 2021) and efficient ranking algorithms (Qin et al., 2023; Zhuang et al., 2024; Chen et al., 2025) aimed at minimizing the number of comparisons while preserving ranking performance.

2.3 Listwise Approaches

Listwise approaches utilize the long-context processing capability of language models by inputting queries along with a candidate document list into the LLM, directly generating ranking results. Existing research primarily falls into two categories: one involves directly generating an ordered list (Pradeep et al., 2023; Sun et al., 2023), while the other is based on probability generation, where the ranking is determined by the log-likelihood values of the first token corresponding to the document labels (Reddy et al., 2024). Listwise approaches can comprehensively capture the differences between documents; however, an increase in input length significantly impacts inference efficiency. To address this issue, current research controls output length through a sliding window design(Pradeep et al., 2023; Sun et al., 2023), optimizing it in conjunction with LLM capabilities. While this method demonstrates strong effectiveness and efficiency, issues such as sensitivity to input order and instability of output remain problematic.

2.4 Setwise Approaches

The setwise approaches directly input query and document sets into LLM to identify the most rele-

Given a query {query}, which of the following two passages is more relevant to the query?
A: {passage(i)}
B: {reference passage(r)}
Output A or B:
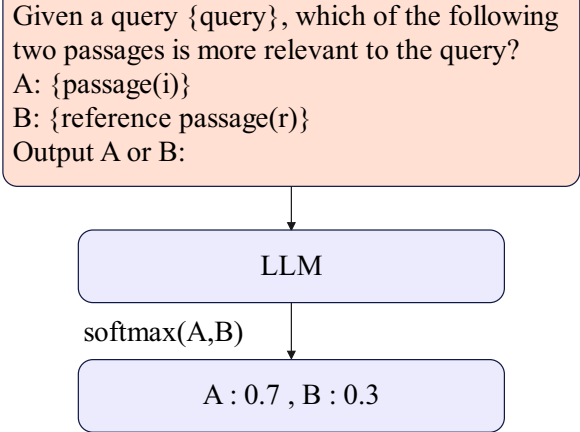
LLM

softmax(A,B)

A : 0.7 , B : 0.3

Figure 2: Reference ranking prompting.

vant document. (Zhuang et al., 2024; Yoon et al., 2024; Podolak et al., 2025). These approaches can be regarded as a formal extension of Pairwise approaches. When employing certain efficient sorting algorithms, such as heap sort and bubble sort, their efficiency significantly exceeds that of Pairwise approaches. Additionally, these approaches can also be seen as a streamlined Listwise approach, effectively achieving a balance between efficiency and effectiveness.

3 RefRank

3.1 Preliminaries

Pointwise approaches treat the documents ranking task as a regression problem. Given a query q and a set of candidate documents $d = (d_1, d_2, \ldots, d_m)$, the LLM ranker takes each query-document pair (q, d_i) as input to assess the relevance of the document to the query ("yes" or "no"). Based on the log-likelihood score $s_{i,0} = \text{LLM}("yes"|q, d_i)$ and $s_{i,1} = \text{LLM}("no"|q, d_i)$, each document score can be computed as follows:

$$f(q, d_i) = \frac{\exp(s_{i,0})}{\exp(s_{i,1}) + \exp(s_{i,0})} \quad (1)$$

Furthermore, candidate documents are ranked based on their document scores. It can be observed that Pointwise approaches use output log-likelihood values as the scoring mechanism. However, these approaches have a significant drawback: the document scoring process is independent, which fails to adequately consider the comparative relevance among documents.

3.2 Reference Ranking Prompting

Building upon the principles of contrastive learning (James, 1980), we implement a Pairwise prompting strategy, as illustrated in Figure 2. The specific execution process is delineated as follows:

Initially, we construct query-reference text pairs (d_i, d_r) to serve as inputs for LLM. This setup guides the LLM in evaluating the relevance of documents based on the prompts provided. LLM outputs are represented as relevance labels for the texts (e.g., A or B). We then normalize the log-likelihoods of outputs A and B using the softmax function, assigning the probability corresponding to A as the final relevance score for the document, denoted as $s(q, d_i, d_r)$. Ultimately, document ranking is conducted based on this relevance score. This ranking method effectively combines the advantages of both Pointwise and Pairwise approaches, leading to improved performance and greater computational efficiency both theoretically and empirically.

3.3 Single Reference Document Selection

The selection of reference documents is a crucial aspect of our method, as it directly influences the quality of the final rankings. Therefore, we conducted an in-depth analytical experiment using the Flan-T5-XL model (Wei et al., 2021) on four datasets: TREC-DL2019 (Craswell et al., 2020), TREC-DL2020 (Craswell et al., 2021), Signal and News (Thakur et al., 2021). NDCG@10 is employed as the evaluation metric for ranking quality in our statistical analysis. All queries are derived from the initial retrieval results obtained through the BM25 method (Lin et al., 2021).

For a given query q, through the first stage of retrieval, recall candidate documents with sorting $d = (d_1, d_2, \ldots, d_m)$. Selecting the j-th as the reference document d_r, we calculate the score for each document $score(d_i, d_r)$. We calculate the ranking quality $N(r)$ based on these scores. Due to varying sorting quality across different datasets, we apply Min-Max normalization to the results of each dataset. The results are presented in Figure 3. It can be observed that as the number of d_i increases, the overall ranking results tend to decline. Therefore, we can obtain the first intentional observation.

$$N(r) = g(\text{sort}(score(d_i, d_r))) \quad (2)$$

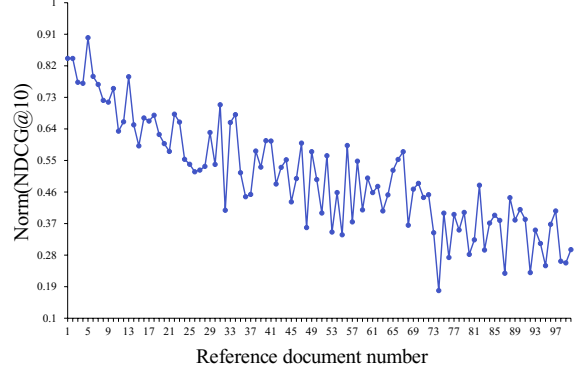for $i = (1, 2, \ldots, m)$, $r \in (1, 2, \ldots, m)$, where $g = \text{NDCG@10}$.

Figure 3: The relationship between reference documents and sorting results.

Observation 1: Selecting the document with the index d_i located earlier as the reference document is more likely to lead to high sorting quality.

To further determine which documents to choose from the index d_j located earlier, we analyze the variation in ranking quality when randomly selecting one of the first k documents as the reference document. The specific ranking quality is calculated as:

$$S(r) = \frac{1}{k} \sum_{r=1}^{k} N(r), r \in (1, 2, \ldots, k) \quad (3)$$

From Figure 4, it can be observed that when selecting a reference document randomly from the first k documents, the likelihood of choosing lower-quality documents as reference increases with the augmentation of k. Consequently, this leads to a deterioration in the overall ranking quality. Analyzing the average trends across the four datasets, a significant peak is evident at $k = 2$. Therefore, we can conclude that it is advisable to randomly select one document from the top-2 as the reference document.

3.4 Multiple Reference Document Ensemble Strategy

Based on the identification of a reference document, each document can be scored, allowing for multiple evaluations of the same document through various reference documents. Therefore, we propose to optimize the final ranking quality of documents by applying appropriate weighting to these scores. In this framework, each reference document functions as a reviewer, while utilizing multiple reference documents represents multiple reviewers assessing the same document. Theoretically, employing
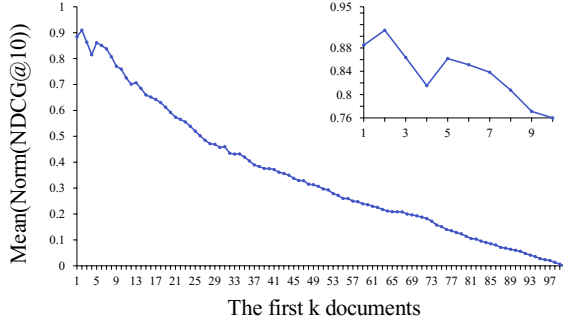
4

Figure 4: The variation in ranking quality when randomly selecting one of the first k documents as the reference document.

Figure 5: The relationship between the weighted quantity of reference document order and the sorting result.

a weighted averaging method can yield more stable and reliable evaluation results. If all reviewers exhibit high scoring accuracy, the aggregated evaluation result after weighting is anticipated to achieve an elevated level of performance.

Given that documents positioned higher in the ranking are more likely to yield high-quality sorting results as reference documents, we conduct a statistical analysis to examine the relationship between the quality of weighted scores $M(r)$ and the number of weighted documents selected in sequence.

$$M(r) = g(\text{sort}(\frac{1}{m}\sum_{r=1}^{m} score(d_i, d_r))) \quad (4)$$

for $r \in (1, 2, \ldots, m)$, where $g = \text{NDCG@10}$.

The findings are illustrated in Figure 5. As the number of sequential weights increases, the quality of rankings shows a significant initial improvement, which then stabilizes at a consistent equilibrium level. From Figure 5, it can be observed that the maximum value is achieved when the number of weights reaches 45. Therefore, we can obtain the second intentional observation.

Observation 2: The sequential weighting of multiple reference documents yields further enhancements in ranking quality. Additionally, it is important to note that an increasing number of weights does not necessarily lead to better outcomes, as there exists an upper limit.

Assuming that we adopt a sequential weighting strategy to select m results for weighting. Given that the average time complexity of the Pairwise comparison-based quicksort algorithm is $O(n \log n)$, we can establish an upper bound for the time complexity of our algorithm at $O(n \log n)$. Our algorithm has a time complexity of $O(mn)$,
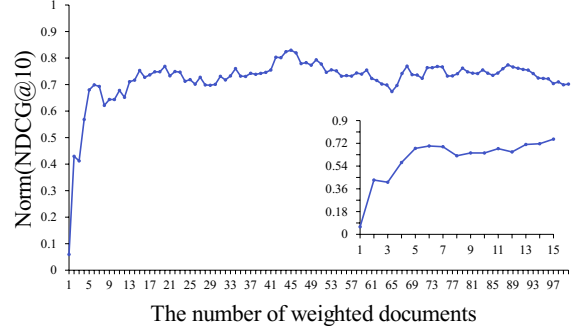
which implies that $m < \log n$. For $n = 100$, this yields $m \leq 6$.

4 Experiments

4.1 Datasets

In order to assess the effectiveness of our research, we utilized several standard evaluation datasets prominent in information retrieval, specifically TREC-DL 2019, TREC-DL 2020, and the BEIR dataset. These datasets facilitate empirical analysis by providing a robust framework for evaluation. To maintain the integrity and consistency of our assessment, all query results were generated using a BM25-based initial retrieval method (Lin et al., 2021). From these results, we selected the top 100 candidate segments for subsequent re-ranking. This methodological approach aligns with current trends in mainstream research (Sun et al., 2023; Ma et al., 2023), thereby reinforcing the relevance of our evaluation framework.

The TREC datasets serve as established benchmarks in the realm of information retrieval. Specifically, we focused on two subsets: TREC-DL 2019, which encompasses 43 queries, and TREC-DL 2020, which comprises 54 queries. Notably, all queries are derived from the MS MARCO v1 corpus, a comprehensive resource containing approximately 8.8 million documents.

In addition to the TREC datasets, we incorporated the BEIR dataset, which encompasses a diverse array of information retrieval tasks across multiple domains. For our analysis, we selected Covid, Touche, DBPedia, SciFact, Signal, News, and Robust04.

The choice of these datasets and subsets enables a nuanced examination of retrieval effectiveness across varied contexts, thereby enhancing the valid-

ity and applicability of our research findings. The principal evaluation metric employed in our study is NDCG@10, which is the official standard metric for the respective datasets. This metric not only facilitates comparability across different studies but also lends additional authority to our experimental results.

4.2 Implementation Details

In our study, we utilized the standard configuration of the Pyserini Python library to generate preliminary BM25 ranking results for all experimental datasets (Lin et al., 2021). In the context of the zero-shot re-ranking task that employs LLMs, we scrutinized existing literature to identify key zero-shot prompting techniques that serve as benchmarks. Subsequently, we conducted a systematic evaluation of two distinct variants of the Flan-t5 model (Wei et al., 2021), including Flan-t5-xl (3B) and Flan-t5-xxl (11B). To better understand the impact of different models, we further studied the currently popular transformer decoder-only LLMs, including Llama 3.1-8b and Qwen2.5-7b. We use the average query latency as the evaluation metric to evaluate efficiency. A single GPU is employed for this assessment, and each query is issued one at a time. The average latency for all queries in the dataset is then calculated. The Pointwise and RefRank methods support batch processing, and we utilize the maximum batch size to optimize GPU memory usage and parallel computation, thereby maximizing efficiency. We carried out the efficiency evaluations on a local GPU workstation equipped with an AMD EPYC 7742 64-Core CPU, a NVIDIA DGX A800 GPU with 80GB of memory.

Pointwise Relevance Generation (RG) (Liang et al., 2022): The RG approach prompts the LLM with a combination of a query and a document, subsequently calculating a ranking score predicated on the binary log-likelihood of relevance (e.g., yes/no). **Pairwise** Allpairs and quick sort: We employed the prompts created by Qin et al (Qin et al., 2023). For quick sort, we choose bubble sort. **Listwise** Listwise employs a List Generation strategy, with the design of the prompts informed by established research. We employed the prompt created by Sun et al (Sun et al., 2023). **Setwise** Setwise is an efficient sorting method that is achieved by selecting the maximum from a set of lists. We employed the prompt created by Zhuang et al (Zhuang et al., 2024).

By categorizing and systematically evaluating

these methods, we aim to provide a comprehensive understanding of the contrasting techniques for zero-shot re-ranking and their implications in information retrieval.

5 Results and Analysis

5.1 Ranking Effectiveness

The control results of our experiment were referenced against the findings of Zhuang et al. (2024) and Qin et al. (2023). Table 1 presents the evaluation results on the TREC-DL dataset. The following findings can be made:

1. The RefRank method demonstrates a significant improvement compared to the Pointwise approach, indicating that introducing reference documents as a comparative benchmark enhances the accuracy of LLM document evaluations.

2. When using a single reference document, the RefRank method underperforms relative to the Pairwise method. However, when multiple weighted approaches are employed, their performance aligns with the Pairwise methods. This further substantiates the effectiveness of incorporating varying reference documents to mitigate the evaluation bias associated with single-reference assessments, leading to an overall enhancement in evaluation performance.

3. In comparison to Listwise and Setwise approaches, the RefRank method with a single-document ranking strategy shows slightly lower performance. Nevertheless, after adopting the reference document integration strategy, the ranking improved, achieving a level comparable to that of Listwise and Setwise methods. This demonstrates the effectiveness of weighting multiple reference documents.

4. An analysis of different model types reveals that the Flan-T5 model, based on an encoder-decoder architecture, outperforms the decode-only architectures, such as Llama3.1-8b and Qwen2.5-7b. This suggests that an encoder-decoder architecture may facilitate a superior comprehension of the query-document pairs in document ranking tasks. Consequently, there is considerable potential for the advancement of encoder-decoder architectures in the context of document ranking.

Table 1: On TREC-DL 2019 and TREC-DL 2020, the overall NDCG@10 achieved by each method is presented. The best results are highlighted in **bold and underlined**, while the second-best results are underlined.

LLM	Methods	DL-19	DL-20
-	BM25	0.506	0.480
Llama3.1-8b	Pointwise-RG	0.601	0.567
	Pairwise-Bubblesort	0.663	0.614
	Listwise	0.652	**0.660**
	Setwise-Heapsort	0.654	0.573
	RefRank-Single(1)	0.669	0.601
	RefRank-Multiple(5)	**0.707**	0.624
Qwen2.5-7b	Pointwise-RG	0.676	0.647
	Pairwise-Bubblesort	0.510	0.481
	Listwise	**0.722**	**0.687**
	Setwise-Heapsort	0.682	0.652
	RefRank-Single(1)	0.683	0.645
	RefRank-Multiple(5)	0.688	0.656
Flan-t5-xl	Pointwise-RG	0.650	0.636
	Pairwise-Bubblesort	0.683	0.662
	Listwise	0.569	0.547
	Setwise-Heapsort	0.693	0.678
	RefRank-Single(1)	0.698	0.659
	RefRank-Multiple(5)	**0.705**	**0.682**
Flan-t5-xxl	Pointwise-RG	0.644	0.632
	Pairwise-Bubblesort	0.671	0.681
	Listwise	0.701	0.690
	Setwise-Heapsort	**0.706**	0.688
	RefRank-Single(1)	**0.706**	0.682
	RefRank-Multiple(5)	0.696	**0.702**

Table 2 displays the evaluation outcomes for the Flan-T5 model on the BEIR dataset. Key findings include:

1. Consistent with previous results on the TREC-DL dataset, the RefRank method universally surpasses the Pointwise approach.

2. The RefRank method performs well across multiple datasets, illustrating the broad applicability of using reference documents as a comparative standard.

3. The performance difference between Flan-T5-XL and Flan-T5-XXL is not significant, suggesting that both models possess comparable capabilities in text ranking tasks. Therefore, in practical applications, we may prioritize the model with fewer parameters.

5.2 Query Latency

Figure 6 shows the average query latency of different methods using the Llama3.1-8b model on TREC-DL 2019. We can obtain the following findings.
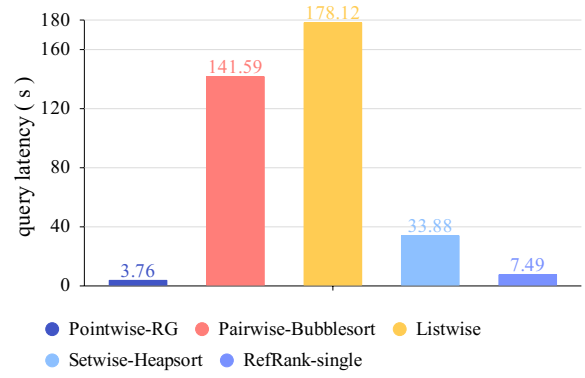

Figure 6: The average query latency of different methods using the Llama 3.1-8b model on TREC-DL 2019.

1. RefRank and Pointwise methods have comparable query latencies, significantly lower than others. This can be attributed to two reasons: first, both the RefRank method and the Pointwise method can obtain document scores in a single inference, allowing for efficiency improvements through batching and parallel computation; second, the input lengths for the RefRank method and the Pointwise method are relatively short, leading to faster inference times.

2. The query latency of the RefRank method is slightly higher than that of the Pointwise method, primarily due to the introduction of reference documents, which increases the input length.

6 Ablation Studies

To systematically evaluate the robustness and effectiveness of the RefRank, we designed and conducted a series of ablation experiments to assess the performance of the framework from various perspectives.

Robustness of Choosing Different Reference Documents. In our experiments, we selected the top-1 document for evaluation. Building on the analysis presented in Section 3.3, we computed the ranking results using the top-2 documents as reference documents. As illustrated in Figure 7, the results on both the TREC-DL and BEIR datasets demonstrate stability, with an average fluctuation of merely 0.34%.

Robustness of Selecting Different Weighted numbers. In this experiment, we utilized the top-3, top-4, and top-5 weighted documents. As shown in Figure 8, the results across the TREC-DL and

Table 2: On the TREC-DL 2019, TREC-DL 2020, and BEIR datasets, the overall NDCG@10 achieved by each method is presented. The best results are highlighted in **bold and underlined**, while the second-best results are underlined.

LLM	methods	COVID	Touche	DBPedia	SciFact	Signal	News	Robust04	Avg
-	BM25	0.595	0.442	0.318	0.679	0.331	0.395	0.407	0.452
Flan-t5-xl	Pointwise-RG	0.698	0.269	0.273	0.553	0.297	0.413	0.479	0.426
	Pointwise-QLM	0.679	0.216	0.310	0.696	0.299	0.422	0.427	0.436
	Pairwise-Bubblesort	0.776	<u>0.405</u>	**0.448**	**0.734**	**0.356**	<u>0.465</u>	0.507	**0.527**
	Pairwise-Allpairs	**0.819**	0.269	<u>0.446</u>	<u>0.733</u>	0.321	<u>0.465</u>	<u>0.540</u>	0.513
	Listwise	0.650	**0.451**	0.366	0.694	<u>0.349</u>	0.437	0.475	0.489
	Setwise-Heapsort	0.757	0.283	0.428	0.677	0.314	<u>0.465</u>	0.520	0.492
	RefRank-Single(1)	0.802	0.279	0.421	0.709	0.292	0.452	0.521	0.496
	RefRank-Multiple(5)	<u>0.818</u>	0.291	0.445	0.714	0.300	**0.499**	<u>0.535</u>	<u>0.515</u>
Flan-t5-xxl	Pointwise-RG	0.691	0.240	0.305	0.623	0.274	0.392	0.515	0.434
	Pointwise-QLM	0.707	0.188	0.324	0.712	0.307	0.431	0.440	0.444
	Pairwise-Bubblesort	0.744	<u>0.416</u>	<u>0.422</u>	0.725	<u>0.351</u>	0.473	0.524	**0.522**
	Pairwise-Allpairs	<u>0.796</u>	0.298	0.414	<u>0.742</u>	0.322	<u>0.477</u>	**0.568**	<u>0.517</u>
	Listwise	0.664	**0.453**	**0.441**	0.736	**0.353**	0.458	0.495	0.514
	Setwise-Heapsort	0.752	0.297	0.402	0.726	0.321	0.473	0.513	0.498
	RefRank-Single(1)	0.783	0.296	0.400	0.739	0.301	0.447	0.531	0.500
	RefRank-Multiple(5)	**0.799**	0.296	0.412	**0.755**	0.310	**0.480**	<u>0.554</u>	0.515

Figure 7: Robustness of Choosing Different Reference Documents.
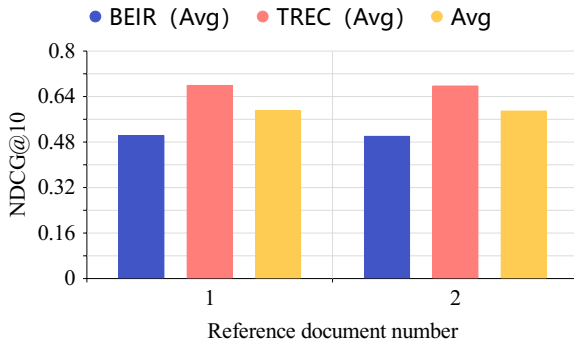

Figure 8: Robustness of Selecting Different Weights.

BEIR datasets remain stable, with an average fluctuation of 0.66%. Therefore, in practice, weighting the top-3 documents can better balance efficiency and effectiveness.

7 Conclusion

In this study, we conduct a systematic analysis of a zero-shot document ranking approach based on LLMs. We innovatively propose **RefRank**: a simple and effective comparative ranking method based on a fixed reference document. The core innovation lies in the introduction of reference documents as comparative benchmarks, which effectively constructs an information comparison mechanism between different documents, ensuring the validity of the ranking system. Using the same evaluation strategy as the Pointwise method en-
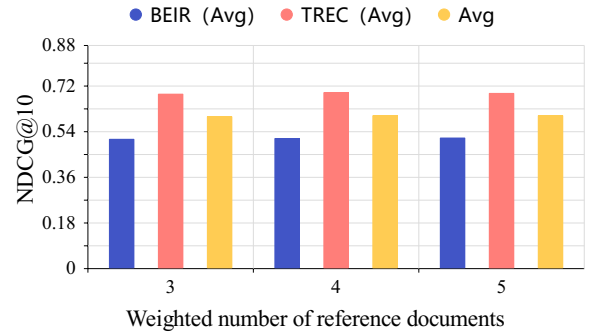
sures efficiency. By leveraging the ranking information from the initial retrieval results, the top two documents are recommended as optimal reference documents. Moreover, inspired by ensemble optimization, we introduce a multiple reference document ensemble strategy to enhance ranking quality. The effectiveness of this method has been validated across various models. Future research may explore expanding reference documents into a comprehensive set and investigate the integration of LLM-generated documents with selected references, both promising avenues for study.

Limitations

This method relies on the log-likelihood values outputted by the model as the foundational metric for document relevance scoring. This technical approach results in the research scope being primarily limited to open-source model ecosystems, as these allow direct access to the output layer's probability distributions. For closed-source models that employ black-box architectures (such as certain commercial APIs), if it is not possible to obtain the complete probability output or log-likelihood values, then there exist adaptability challenges in the technical implementation of this method. This limitation may affect the universal applicability of the method in industrial-grade retrieval systems, and future research must explore compatibility solutions based on confidence estimation or alternative interpretable features.

References

M Agrawal, S Hegselmann, H Lang, Y Kim, and D Sontag. 2023. Large language models are zero-shot clinical information extractors. arxiv, 2022. *arXiv preprint arXiv:2205.12689*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yiqun Chen, Qi Liu, Yi Zhang, Weiwei Sun, Xinyu Ma, Wei Yang, Daiting Shi, Jiaxin Mao, and Dawei Yin. 2025. Tourrank: Utilizing large language models for documents ranking with a tournament-inspired strategy. In *Proceedings of the ACM on Web Conference 2025*, pages 1638–1652.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track. *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*.

Lukas Gienapp, Maik Fröbe, Matthias Hagen, and Martin Potthast. 2022. Sparse pairwise re-ranking with pre-trained transformers. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 72–80.

Fang Guo, Wenyu Li, Honglei Zhuang, Yun Luo, Yafu Li, Le Yan, Qi Zhu, and Yue Zhang. 2025. Mcranker: Generating diverse criteria on-the-fly to improve pointwise llm rankers. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 944–953.

Carl James. 1980. Contrastive analysis.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

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*, pages 2356–2362.

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*.

Aliaksei Mikhailiuk, Clifford Wilmot, Maria Perez-Ortiz, Dingcheng Yue, and Rafał K Mantiuk. 2021. Active sampling for pairwise comparisons via approximate message passing and information gain maximization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2559–2566. IEEE.

Jakub Podolak, Leon Peric, Mina Janicijevic, and Roxana Petcu. 2025. Beyond reproducibility: Advancing zero-shot llm reranking efficiency with setwise insertion. *Preprint*, arXiv:2504.10509.

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

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, and 1 others. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563*.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. First: Faster improved listwise

reranking with single token decoding. *arXiv preprint arXiv:2406.15657*.

Noelia Rivera-Garrido, María del Pino Ramos-Sosa, Michela Accerenzi, and Pablo Brañas-Garza. 2022. Continuous and binary sets of responses differ in the field. *Scientific reports*, 12(1):14376.

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. *arXiv preprint arXiv:2204.07496*.

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. *arXiv preprint arXiv:2304.09542*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Shuai Wang, Harrisen Scells, Bevan Koopman, and Guido Zuccon. 2023. Can chatgpt write a good boolean query for systematic review literature search? In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 1426–1436.

Jason Wei, Maarten Bosma, Vincent Y 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. *arXiv preprint arXiv:2109.01652*.

Jinxi Xu and W. Bruce Croft. 2017. Quary expansion using local and global document analysis. *SIGIR Forum*, 51(2):168–175.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on web search and data mining*, pages 1154–1156.

Soyoung Yoon, Eunbi Choi, Jiyeon Kim, Hyeongu Yun, Yireun Kim, and Seung-won Hwang. 2024. Listt5: Listwise reranking with fusion-in-decoder improves zero-shot retrieval. *arXiv preprint arXiv:2402.15838*.

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. *arXiv preprint arXiv:2310.14122*.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023b. Open-source large language models are strong zero-shot query likelihood models for document ranking. *arXiv preprint arXiv:2310.13243*.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47.