MCRanker: Generating Diverse Criteria On-the-Fly to Improve Pointwise LLM Rankers

Fang Guo*
Zhejiang University
Hangzhou, China
guofang@westlake.edu.cn

Wenyu Li*
South China University of Technology
Guangzhou, China
wenyulilwy@gmail.com

Honglei Zhuang
Google
Seattle, USA
hlz@google.com

Yun Luo
Westlake University
Hangzhou, China
luoyun@westlake.edu.cn

Yafu Li
Westlake University
Hangzhou, China
liyafu@westlake.edu.cn

Le Yan
Google
Mountain View, USA
lyyanle@google.com

Qi Zhu
Zhejiang University
Hangzhou, China
qizhu.zju.research@gmail.com

Yue Zhang
Westlake University
Hangzhou, China
zhangyue@westlake.edu.cn

Abstract

The most recent pointwise Large Language Model (LLM) rankers have achieved remarkable ranking results. However, these rankers are hindered by two major drawbacks: (1) they fail to follow a standardized comparison guidance during the ranking process, and (2) they struggle with comprehensive considerations when dealing with diverse semantics of the query and complicated info in the passages. To address these shortcomings, we propose to build a zero-shot pointwise ranker that first recruits a virtual annotation team to generate query-based criteria from various perspectives and then uses these criteria to conduct an ensemble passage evaluation. Additionally, we are among the first to explore how criteria can be generated automatically and used in text ranking tasks. Our method, tested on eight datasets from the BEIR benchmark, demonstrates that incorporating this multi-perspective criteria ensemble approach significantly enhanced the performance of pointwise LLM rankers. [1]

CCS Concepts

• **Information systems** → **Learning to rank**.

Keywords

Pointwise Ranking, Text Ranking, LLM Ranking, Agent

ACM Reference Format:
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 (WSDM '25), March 10–14, 2025, Hannover, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3701551.3703583

*Both authors contributed equally to the paper
[1]Code, data, and prompts can be found at: please click to visit our github page

WSDM '25, March 10–14, 2025, Hannover, Germany

ACM ISBN 979-8-4007-1329-3/25/03
https://doi.org/10.1145/3701551.3703583

1 Introduction

The integration of Large Language Models (LLMs) into various text rankers has resulted in notable advancements, outperforming traditional neural ranking approaches even in a zero-shot setup [13, 28, 30, 37, 56]. In the realm of LLM rankers, pointwise rankers are one of the major categories that evaluate each passage individually without comparing its position or relation to other passages [53]. Compared to pairwise and listwise rankers, pointwise rankers[17, 33, 55] own its advantage in lower token cost, ease of deployment in practical applications, and stronger interpretability.

Nevertheless, the iterative nature of querying LLMs, combined with their stochastic behavior, leads to inconsistent assessment criteria for zero-shot pointwise rankers, resulting in a lack of both *consistency* and *comprehensiveness* in the ranking process. Figure 1 illustrates such a concrete example in orange straight line. The inconsistency happens when the contrasting scores are assigned to content with similar semantics. For instance, both Document 1 and Document 2 discuss the comparative effectiveness of various mask types in preventing COVID-19 transmission but are assigned markedly different scores by the pointwise ranker. Moreover, Document 0, which is tangentially related to the query erroneously receives a high score. This error happens when the LLM ranker adopts a biased assessment criterion that prioritizes keyword presence over a subtle understanding of content semantics. Consequently, an optimal pointwise ranker should mitigate the influence of the LLM's stochastic behavior and effectively manage nuanced query-passage relevance.

In this paper, we investigate how to build a zero-shot pointwise ranker that is capable of generating both consistent and comprehensive assessments of passages. Inspired by how professional human annotators work, we approach the query-passage pair evaluation process of a pointwise ranker as a virtual annotation process. Recent studies on human annotation practices [8, 40] suggest that optimal

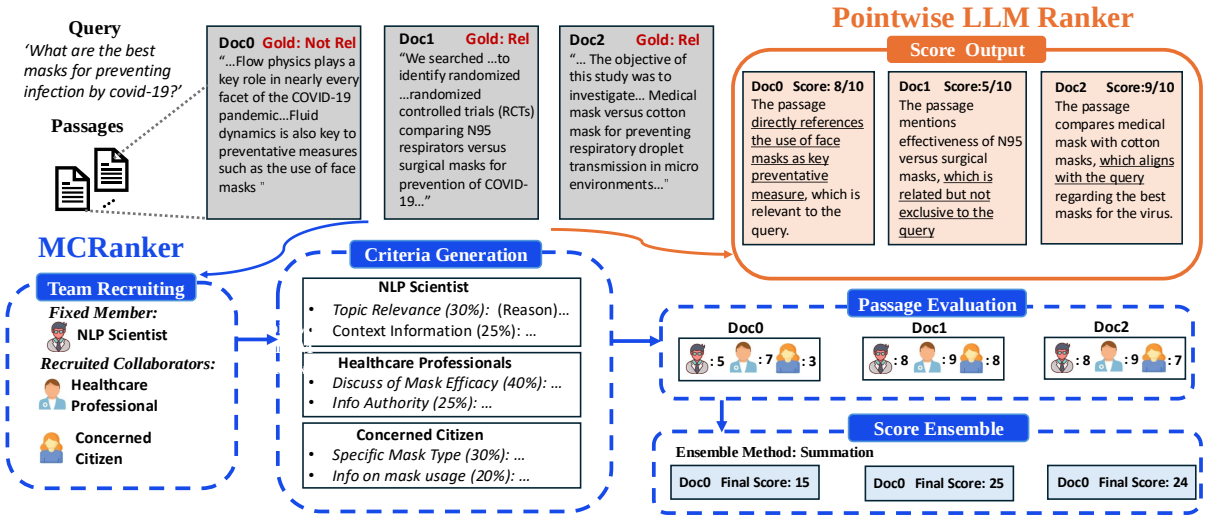

Figure 1: Pipeline of the proposed MCRanker in blue dashed line and the example output of the Pointwise LLM-based Ranker in orange straight line.

annotation outcomes are achieved by thoroughly considering the comprehensive semantics of the query, bolstered by standardized annotation criteria. Given the unknown perspective of the individual posing the query, the collaboration between domain-specific experts and language experts is essential to address the semantic diversity within the query. For example, when faced with the query "*What are the best masks for preventing infection by COVID-19?*", a healthcare professional might prioritize passages comparing different mask types, whereas an NLP scientist might focus more on the linguistic features of the passage. Moreover, clear or well-defined annotation criteria can significantly enhance the quality of the annotation process [8, 41]. Therefore, it is crucial to establish precise and well-articulated criteria to guide the annotation process.

In line with these guidelines for the human annotation process, we introduce the MCRanker framework, which generates query-based **M**ulti-perspective **C**riteria to improve zero-shot pointwise LLM **Ranker**. To provide a *comprehensive* assessment of query-passage relevance, the MCRanker framework draws inspiration from recent work that adopts the "Multi-Perspective Problem Solving" philosophy [7, 14, 18, 25, 46]. In particular, we design a *Team Recruiting* module that simulates domain expertise and text analytical capabilities akin to human query analysis by assembling a virtual annotation team. To ensure *consistent* passage assessments, MCRanker's *Criteria Generation* module enables each team member to develop query-specific criteria to guide the passage evaluation. Unlike previous work in generative evaluation that relies on handcrafted criteria [20, 21, 39], our criteria generation process is fully automated. Figure 1 illustrates the pipeline of MCRanker for query "*What are the best masks for preventing infection by COVID-19?*", highlighted in blue dashed line. After reading the query, MCRanker first builds a virtual annotation team. This team comprises a fixed "NLP scientist" and recruits two collaborators: a "health professional" and a "concerned citizen", each contributing their unique perspective to the annotation process. Upon reviewing the query, each team member generates a set of weighted criteria reflective of their viewpoints. For example, the NLP scientist emphasizes text analysis criteria such as "topic relevance" and "keywords match", while the health professional focuses on the specificity of mask-related information. In the passage evaluation phase, each team member independently assigns a score to the passage based on his established criteria. Finally, individual assessments are aggregated into an ensemble score that reflects the passage's relevance to the given query.

In summary, the key contributions of this paper are as follows:

- We introduce MCRanker, a pointwise ranker that incorporates a multi-perspective framework to generate query-level criteria. These criteria then guide the evaluation of query-passage pairs.
- We investigate how criteria can be generated automatically and further used to influence a ranker. The in-depth analysis reveals the necessity of query-centric criteria to enhance ranking performance.
- We evaluate MCRanker on 8 datasets from the BEIR benchmark. The results demonstrate that our proposed approach consistently delivers superior ranking results across various datasets.

2 Related Work

Text Ranking Recent studies have been investigating the application of large language models (LLMs) for zero-shot text ranking [9, 19, 26, 27, 31, 47, 50, 54, 56]. Notably, pairwise [30] and listwise [23, 28, 29, 37, 57] LLM rankers involve the simultaneous evaluation of two or more documents to generate a ranked list. However, these ranking strategies typically necessitate the establishment of a quality initial document order, which is often provided by a first-stage ranker, such as a pointwise ranker.

Pointwise rankers evaluate each query-passage pair separately, offering benefits in terms of scalability and interpretability. Several pointwise rankers derive the relevance score from the likelihood of

the document's relevance to the query [17] or the probability of generating the query from the document [33]. Zhuang et al. leverages fine-grained relevance labels within prompts to enable more nuanced differentiation among documents. Nevertheless, these models face consistency issues and cannot make comprehensive passage assessments.

Prompt-based Generative Evaluation Amid the rapid development of large language models (LLMs), an emerging body of research has increasingly focused on utilizing these models as evaluators for natural language generation (NLG) tasks. Within this research, prompt-based evaluation methods typically involve prompting LLMs to assess generated text through tailored prompt engineering and criteria construction [6, 11, 16, 21]. G-Eval [20] introduced a prompt-based evaluator that relies on customized, human-curated evaluation criteria for different NLG tasks. Liu et al. builds scoring criteria by first annotating human expert labels and then allowing the LLM to draft and self-refine the criteria based on these labels.

In contrast, MCRanker's query-specific criteria are automatically crafted by a virtual annotation team. Moreover, to the best of our knowledge, we are among the first to explore the significance, utility, and generalizability of these criteria in the field of text ranking.

Zero-shot LLM Assessors Recent studies have explored the LLMs' capability as pseudo-assessors [4, 5, 32, 39, 42]. The primary objective of these LLM assessors is to assign a relevance label to each query-passage pair. Ideally, these relevance labels should be in line with human-generated ground-truth relevance labels. For example, Thomas et al. designed prompts that incorporate several handcrafted aspects while Wang et al. have introduced a pipeline to let LLMs perform expert-level dataset annotation.

LLM assessors are built to establish evaluation datasets, marking a distinct departure from the goal of LLM rankers. LLM rankers primarily focus on ensuring the accuracy of the relative ordering among the top-ranked documents, without necessarily providing explicit relevance labels for each query-passage pair.

Multi-Perspective Systems The concept of "Multi-Perspective Problem Solving" has garnered enormous attention in recent research, particularly within the domains of "Multi-Agent" systems [3, 7, 14, 18, 25, 44, 46, 48] and "Mixture-of-Experts" systems [15], demonstrating its potential in resolving complex tasks. Notably, recent advancements in multi-agent systems have demonstrated the capacity of LLMs to assume specific identities, either through automatic assignment [34, 45] or manual selection [12, 49, 51]. These identified LLM agents follow various collaboration mechanisms [2, 46, 52] to accomplish given tasks like video creation [35], dramatic scenario simulation [24], dialogue generation [1, 43] and medical report generation [36].

Inspired by the similar philosophy, we designed a "Team Recruiting" module, which automatically generates a few collaborators to work with a fixed NLP scientist. As far as we know, we are among the first to adopt this idea to solve a text ranking problem. We did not study the collaboration mechanism within the team and left it for future work.

3 Methods

3.1 Preliminary

We formally describe how a zero-shot pointwise LLM ranker tackles a ranking problem. Given a specific query denoted as q, along with a set of passages to be evaluated as $P = (p_1, \ldots, p_m)$. The pointwise ranking function, represented as f, evaluates each pair consisting of the query q and an individual passage p_i. It computes a score $s_i = f(q, p_i)$, signifying each passage's relevance to the query. After the pointwise ranker has computed relevance scores for all query-passage pairs (q, p_i), it ranks all passages P based on predicted scores $S = (s_1, \ldots, s_m)$ in descending order and outputs the ranked list as the final result. Notice that zero-shot pointwise ranker can be subdivided into likelihood-based and text-based. Likelihood-based methods derive ranking scores from the generative likelihood of LLMs, whereas text-based methods derive their ranking scores directly from the textual outputs of the LLMs. In this study, we concentrate on the text-based pointwise ranker because token likelihood is in general unavailable in closed-source LLMs like GPT. However, likelihood-based LLM rankers can be seamlessly plugged into our proposed framework.

3.2 Modules

To ensure that a pointwise ranker provides consistent and comprehensive query-based passage evaluations using automatically generated criteria, we propose MCRanker which instructs LLM to perform the following modules: (1) **Query-Based Team Recruiting** that recruits a virtual annotation team based on the query. This team includes a designated NLP scientist and a few collaborators with different domain expertise to provide various viewpoints for the upcoming annotation step; (2) **Criteria Generation** that prompts each team member to create detailed criteria for the upcoming evaluation step; (3) **Passage Evaluation** that lets each team member evaluate query-passage pairs following his criteria and (4) **Score Ensemble and Ranking** that ranks on the final scores, which are ensemble of team members' evaluation results. Figure 1 shows a working example of MCRanker during inference.

Query-based Team Recruiting. The first step is to establish a virtual team endowed with expert-level annotation capabilities, encompassing domain-specific expertise and text analysis proficiency. To acquire domain knowledge, we prompt a team recruiting LLM $M_{Recruit}$ to let it decide each team member's identity. The prompt is designed to ask $M_{Recruit}$ to guess who comes out of the given query. To ensure a comprehensive skill set across the team, the prompt fosters an interdisciplinary mix of talents. In alignment with our analysis of the human annotation process in the preceding "Introduction" section, our virtual annotation team also necessitates advanced text analysis skills. However, we empirically find that $M_{Recruit}$ always fails to recruit a professional in text and language. Therefore, we designate an NLP scientist to join the team. The fully formed annotation team A is made up of an NLP scientist a_0 and other recruited collaborators $a_1, \ldots, a_n$. This recruiting process can be described as:

$$a_1, \ldots, a_n = M_{Recruit}(x_r(q)) \tag{1}$$

where x_r is the prompt for team recruiting that takes the query as input.

Criteria Generation. In the process of professional annotation, adherence to established scoring criteria is crucial for annotators to maintain uniformity in their evaluations. In light of this, our virtual annotation team mandates that each member formulate their own set of scoring criteria. These query-centric criteria are also expected to include a weighted distribution for each criterion, ensuring a systematic assessment. This step is formulated as:

$$c_j = M_{Criteria}(x_c(q, a_j)) \tag{2}$$

where $0 \leqq j \leqq n$, $M_{Criteria}$ is the LLM responsible for criteria generation. x_c is the corresponding prompt for $M_{Criteria}$.

Passage Evaluation. The member of the annotation team then starts to evaluate the passages. For each query-passage pair (q, p_i), the evaluation is fulfilled through prompting a passage evaluation LLM $M_{Evaluation}$. It processes by letting each team member read the query, the passage as well as the query-centric criteria that he established in the previous step. Then each team member is expected to rate the relevance on a scale from 0 to k. This evaluation procedure can be described as:

$$s_{ij} = M_{Evaluation}(x_e(q, p_i, a_j, c_j)) \tag{3}$$

where x_e is the prompt for passage evaluation.

Score Ensemble. Once we get the passage evaluation result from each team member, we can ensemble their result to get a final score s_i. We consider three different ensemble methods: (1) Score Summation, (2) Reciprocal Rank, and (3) LLM Assessor $M_{assessor}$.

For (1) Score Summation, we obtain the final score of a passage by simply adding up the scores from each team member. For (2) Reciprocal Rank, we first rank each team member a_j's evaluation result to get a ranked list r^j. Then we calculate each passage's final score by summing up its mean reciprocal rank scores in each ranked list. For (3) LLM Assessor, we prompt $M_{Assessor}$ by feeding in each team member's evaluation result on the passage and letting it give an overall score.

The three computing equations are shown below in order:

$$s_{\text{Sum},i} = \sum_{j=0}^{|A|} s_{ij} \qquad s_{\text{RR},i} = \sum_{j=0}^{|A|} \frac{1}{r_i^j} \tag{4}$$

$$s_{\text{Assessor},i} = M_{Assessor}(x_a(q, p_i, s_{i0}, \ldots, s_{ij}, r_{i0}, \ldots, r_{ij}))$$

where x_a is the prompt for the final score assessment. The final output of MCRanker is a ranked list on s_i in descending order.

4 Experimental Setup

4.1 Dataset

Same as [55], our experiments were conducted on 8 datasets from BEIR benchmark [38]: Covid, Touche, DBPedia, SciFact, Signal, News, Robust04, and NFCorpus.

4.2 Compared Methods

We compared various zero-shot pointwise rankers as follows:

(1) **Query Generation (QG) [33]**: This method rescores retrieved passages with a zero-shot question generation model. It uses a pre-trained language model to compute the probability of the input question conditioned on a retrieved passage.
(2) **RankLLaMA [22]**: This pointwise ranker is fine-tuned on MS MARCO dataset using LLaMA-2-13B initialization. It is trained to process a query and a candidate document together as model input and generates a relevance score from the representation of the last token.
(3) **Rating Scale Relevance Generation (RG-S) [55]**: This method prompts the LLM to first output the relevance label for each query-passage pair, then calculates the expected relevance value using relevance value and corresponding marginal probability.
(4) **Rating Scale 0-to-k Directly Score (DIRECT(0, k))**: This method prompts the LLM to directly generate the relevance score for each query-passage pair. We adopt the prompt from Zhuang et al. and k represents the rating scale.

Note that QG and RG-S are all likelihood-based pointwise rankers, while DIRECT(0, k) and MCRanker are text-based. Likelihood-based models require the base LLM to have access to log probabilities for arbitrary tokens, which prevents us from running them on GPT-4. Consequently, we directly reported their performance as presented in the original papers.

Additionally, a more comprehensive comparison with pairwise and listwise ranking methods will be provided in Supplementary Material. However, since our primary focus is on pointwise rankers, comparisons with these methods fall outside the scope of this study.

4.3 Configurations

We first used BM25 through pyserini[2] to retrieve the top-100 documents from each query of every dataset, then ranked the retrieved documents with our MCRanker and the baseline methods. The ranking performance was measured by NDCG@10 [10].

For MCRanker, we used "GPT-4-1106-Preview" as the base LLM model for $M_{Recruit}$, $M_{Criteria}$, $M_{Evaluation}$ and $M_{Assessor}$. The temperature is set to 0 and the rating scale k is set to 10. For DIRECT(0, k), "GPT-4-1106-Preview" served as the base LLM with the temperature similarly set to 0. For $M_{Evaluation}$, we prompt it to output a score without further explaining the reason, this setup is in line with RG-S. The default MCRanker incorporates a virtual annotation team comprising an NLP scientist and two additional collaborators. The variants of the annotation team are indicated by the actual member identities. For example, NLP Sci. represents the NLP Scientist, R.C. represents the Recruited Collaborator, and MCRanker$_{NLPSci.+R.C.}$ means the annotation team in this variant has one NLP scientist and one recruited collaborator.

The ensemble mechanism used for MCRanker was set to "Score Summation" by default, namely the first equation in Equation 4. When needed, the method name was appended with the suffix "-RE" if calculated by Rank Ensemble, and "-$M_{Assessor}$" by the LLM assessor.

5 Results

5.1 Overall Performance

Table 1 summarizes the overall performance on eight datasets from BEIR. Our method achieves the best performance and outperforms

[2]https://github.com/castorini/pyserini

Table 1: Overall ranking performances measured by NDCG@10 on BEIR. "*" in Method means likelihood-based models that adopt the token probability for relevance score calculation. The best performances are bold and underlined, and the second is underlined.

Method	Model	Covid	Touche	News	Signal	DBPedia	SciFact	Robust04	NFCorpus	Avg
BM25	N/A	59.47	**44.22**	39.52	**33.05**	31.80	67.89	40.70	30.75	43.42
QG*	FLAN PaLM2 S	73.57	24.08	41.56	28.72	37.73	74.95	46.51	36.73	45.48
RankLLaMA*	LLaMA-2	**85.2**	40.1	46.67	29.87	**48.3**	73.2	46.97	30.3	50.08
RG-S*	FLAN PaLM2 S	80.48	27.57	47.90	33.01	41.90	**75.21**	56.68	39.01	50.22
DIRECT(0, 10)	GPT-4-1106-Preview	79.30	25.22	46.19	29.12	40.82	70.08	53.78	37.52	47.75
DIRECT(0, 20)	GPT-4-1106-Preview	79.96	22.05	47.57	27.77	40.53	70.53	54.66	37.19	47.53
MCRanker$_{NLPSci.}$	GPT-4-1106-Preview	81.49	29.33	46.13	29.48	41.25	70.86	56.37	38.25	49.14
MCRanker$_{R.C.}$	GPT-4-1106-Preview	82.43	30.60	48.52	26.58	41.11	71.35	55.78	37.93	49.28
MCRanker$_{2R.C.}$	GPT-4-1106-Preview	83.53	31.42	**50.90**	26.85	42.33	71.86	56.84	38.36	50.26
MCRanker$_{NLPSci.+R.C.}$	GPT-4-1106-Preview	84.16	32.91	49.54	29.94	43.85	73.33	57.12	39.12	51.24
MCRanker	GPT-4-1106-Preview	84.23	32.48	50.32	29.73	44.67	73.14	**57.23**	**39.58**	**51.42**

the direct GPT4-prompting ranker DIRECT(0, 10) in average performance by nearly 8% in NDCG@10.

Upon analyzing the table, several observations can be inferred: (1) The method introduced in this study, MCRanker, consistently outshines the baseline across all datasets. When compared with the direct score baseline DIRECT, MCRanker displays a remarkable improvement, registering an average increase in NDCG@10 by a magnitude of 3.67. This enhancement underscores the efficacy of MCRanker in augmenting the LLM's capability for more accurate relevance predictions. (2) The default version MCRanker that recruited two collaborators besides an NLP scientist shows improvements in performance over the single-collaborator variant MCRanker$_{NLPSci.+R.C.}$ in five datasets, although this improvement is not statistically substantial. This may be attributed to the overlapping expertise between the NLP scientist and the other collaborators. A more granular analysis of the multi-perspective annotation will be provided in Section 5.2.1. (3) A comparative examination of the results from DIRECT with rating intervals of $(0, 10)$ and $(0, 20)$ reveals that simply expanding the scale does not enhance the LLM's ability to discriminate between relevant and non-relevant passages. This finding further demonstrates the performance gain observed in MCRanker can be ascribed to the ensemble evaluation from each perspective. (4) MCRanker achieves higher scores than likelihood-based ranker RG-S in six out of the eight datasets, except the Signal and SciFact datasets. We hypothesize that the brevity of the Twitter posts in Signal and the ambiguous query-passage relevance in SciFact may result in a confounding effect on the multi-perspective criteria evaluation, thereby impairing MCRanker's performance. Also, we believe that if we could get arbitrary token probability from GPT4, the performance of MCRanker can be further enhanced.

5.2 Ablation Study

5.2.1 Study on multi-perspective annotation. The virtual annotation team in the MCRanker framework includes a designated NLP scientist and a few recruited collaborators. Our experiments involved different combinations of team members to evaluate the efficacy of our approach.

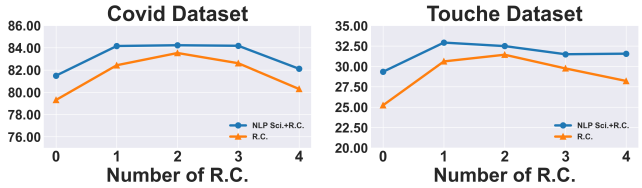

Figure 2: Study on different number of Team Member.

As shown in table 1, the inclusion of only one team member like one NLP scientist variant MCRanker$_{NLPSci.}$ or one recruited collaborator variant MCRanker$_{R.C.}$ contributes to a performance improvement when compared with the direct prompt baseline DIRECT(0,k). This suggests that anchoring the annotation criteria to a specific perspective yields more consistent and accurate predictions. Moreover, two recruited collaborators variant MCRanker$_{2R.C.}$ consistently overperform MCRanker$_{R.C.}$ across all eight datasets. This finding supports the notion that an extra perspective can provide a more holistic evaluation, leading to performance gains. While the NLP scientist alone variant MCRanker$_{NLPSci.}$ achieves performance comparable to the single recruited collaborator variant MCRanker$_{R.C.}$ on average, a notable performance gain is observed when the MCRanker$_{NLPSci.+R.C.}$ model is compared to the MCRanker$_{2R.C.}$ variant. We believe it is due to the language model's preference to select annotators with expertise closely aligned with the query, even when prompted to consider interdisciplinary backgrounds. An NLP scientist undoubtedly contributes essential and valuable text analysis skills to the annotation team.

A further experiment was conducted to examine the impact of varying the number of team members. As shown in Figure 2, on both datasets, as the number of R.C. increases, the optimal performance is reached when this number is 2, then the performance starts to drop. This decline might be due to redundancy in expertise among an increased number of collaborators, which introduces noise in the generated criteria from multiple team members. This experiment highlights the importance of determining the number of team members to ensure the optimal ensemble effect within the

Table 2: Ablation study on criteria utility.

Method	Covid	Touche	News
DIRECT(0, 10)	79.30	25.22	46.19
DIRECT(0, 10)+criteria	79.86	30.46	50.22
MCRanker$_{w.o.criteria}$	79.97	26.22	48.00
MCRanker	**84.23**	**32.48**	**50.32**

team. We leave a deeper investigation of the underlying causes and methods to optimize this hyper-parameter for future study.

Table 3: Ablation study on broad criteria. "DBC" represent "Dataset-Based-Criteria".

Method	Covid	Touche	News
DIRECT(0, 10)	79.30	25.22	46.19
MCRanker$_{DBC}$	82.28	27.78	42.63
MCRanker	**84.23**	**32.48**	**50.32**

5.2.2 Study on criteria generation and utility. To assess the impact of criteria on model performance, we employed a systematic ablation study. This involves integrating criteria developed by the virtual annotation team into DIRECT(0,10), removing all criteria from MCRanker, and transitioning from query-based to dataset-based criteria.

Our findings, detailed in Table 2, reveal that stripping MCRanker of its criteria results in a marked performance decline, with decreases of 4.26, 6.26, and 2.32 of NDCG@10 respectively. These results underscore the critical role criteria play in not only guiding rankers in score generation but also ensuring consistency across evaluations. Absent these criteria, scores tend to decrease, reflecting a lack of standardization can easily lead LLM to adopt inconsistent scoring guidelines. What is more, we can observe that incorporating the same criteria used by MCRanker into DIRECT(0,10) yields improvements on all three datasets, with an especially significant enhancement of approximately 5.24 of NDCG@10 on Touche and 4.03 of NDCG@10 on News. This indicates that given criteria as context, the straightforward method DIRECT(0,10), which prompts LLM to generate a relevance score for each query-passage pair can also be enhanced.

Additionally, our analysis also investigates the efficacy of broad criteria. We keep all other steps the same in MCRanker except shifting from query-based to dataset-based criteria in $M_{Criteria}$. The dataset-based criteria are generated by using the dataset description [38] as the query to generate criteria. Under this setting, all queries in one dataset share the same criteria. As Table 3 demonstrates, MCRanker$_{DBC}$ ("DBC" represents "Dataset-Based-Criteria") which adopts a dataset-level criteria has a performance decrease in approximately 1.95 of NDCG@10 for Covid, 4.70 for Touche and 7.69 for News. This decline illustrates the inherent complexity within the benchmark, where queries in a single dataset also diverge in semantics. It also suggests that a one-size-fits-all approach to criteria generation may not achieve optimal prediction performance and query-centric criteria generation step is necessary. Despite this decrease, on dataset Covid and Touche, MCRanker$_{DBC}$ still records a notable performance improvement in comparison to DIRECT(0,10), demonstrating the importance of establishing criteria before scoring. However, for the News dataset, there is a significant performance drop when switching from query-based to dataset-based criteria. This may be attributed to the greater topic diversity within the News dataset, where a dataset-based approach could introduce misleading signals during the query-passage pair evaluation process. Our exploration can shed light on how to balance budget, specificity, and generality in criteria generation.

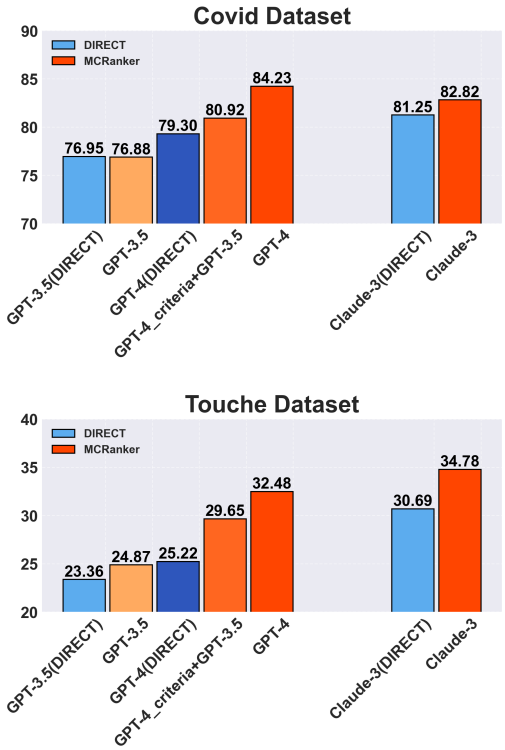

Figure 3: Comparing performance of different base models.

5.3 Generalizability to different LLMs

To explore the generalizability of our proposed methods in different LLMs, we conducted experiments in which we replaced the base LLM with "GPT-3.5-Turbo" and "Claude-3-Sonnet". The result is presented in Figure 3. When using "GPT-3.5-Turbo" for all three modules: $M_{Recruit}$, $M_{Criteria}$, and $M_{Evaluation}$, the performance decreases 7.35 in NDCG@10 on the Covid dataset and 7.61 on the Touche dataset. This level of performance is akin to using "GPT-3.5-Turbo" in a direct prompting baseline DIRECT(0,10). However, maintaining the "GPT-4-1106-Preview" model for $M_{Recruit}$ and $M_{Criteria}$, while only employing "GPT-3.5-Turbo" for $M_{Evaluation}$, results in a substantial performance increase. This finding demonstrates that quality criteria can effectively guide a less advanced $M_{Evaluation}$ to yield more accurate relevance predictions. Upon comparison between the "Claude-3-Sonnet" variant of MCRanker and the corresponding DIRECT(0,k) baseline, it is evident to see the performance increase. These results underscore the effectiveness of high-quality criteria and the capability of large language models

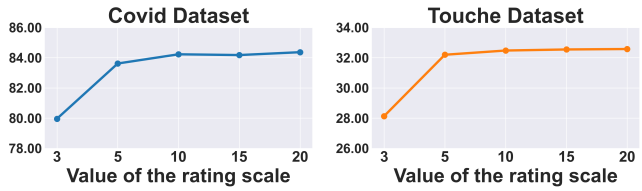

Figure 4: MCRanker with different values of k.

to follow rules set by more advanced ones. Furthermore, they highlight the robust generalizability of our proposed multi-perspective criteria ensemble methodology.

5.4 Impact of Identity and Diversity in the Virtual Annotation Team

Table 4: Ablation study on more team recruiting methods.

Method	Covid	Touche	News
$Random_{R.C.}$	73.95	23.13	19.64
$\text{MCRanker}_{NLPSci.}$	81.49	29.33	46.13
$\text{MCRanker}_{3NLPSci.}$	82.18	29.50	46.82
MCRanker	**84.23**	**32.48**	**50.32**

As shown in Table 4, we conducted an ablation study to verify the effectiveness of the Team Recruiting phase and to emphasize the importance of team member identity. We created a pool of identities by aggregating all identities generated across these three datasets. During each team recruiting process, we randomly selected two identities from the pool with one fixed NLP scientist. This variant of MCRanker is indicated as $Random_{R.C.}$ in the table.

We can observe a significant performance drop in the $Random_{R.C.}$ model. This notable decline underscores the critical importance of forming a query-centric virtual annotation team. Additionally, we conducted another experiment to examine the influence of identity diversity. We designated three NLP scientists in the virtual annotation team, and the results showed a noticeable decrease across all three datasets when compared to the default MCRanker. From this, we can conclude that merely ensembling identical identities does not improve performance; the major performance boost comes from the diversity of the identities.

5.5 Further Analysis

5.5.1 Study on ensemble mechanism. We examine the effect of various ensemble strategies on the performance of MCRanker. The results, detailed in Table 5, indicate that the straightforward "Score Summation" approach surpasses alternative methods on datasets Covid, Touche and News. The limited effectiveness of the "Reciprocal Rank" method might be attributed to the homogeneity in scale of the scores derived from different team members. The LLM assessor, on the other hand, can be easily misled by certain members' scores, which requests for more delicate prompt engineering. Thus, we choose "Score Summation" as the principal ensemble method for its simplicity and robust performance.

Table 5: Ablation study on ensemble mechanism. "R.R." represents "Reciprocal Rank".

Method	Covid	Touche	News
$\text{MCRanker}_{R.R.}$	82.55	30.73	48.19
$\text{MCRanker}_{M_{Assessor}}$	83.58	31.35	49.25
MCRanker	**84.23**	**32.48**	**50.32**

5.5.2 Study on the variance of rating scale. We plot how the performance changes about the rating scale for our proposed MCRanker in Figure 4. On Covid and Touche datasets, when the rating scale k increases from 3 to 5, we can observe an apparent improvement. Then the performance continues to increase as k increases from 5 to 10 and becomes even as k reaches 15 and 20. These observations suggest that our methodology exhibits robustness to variations in different rating scales k when k is chosen within a reasonably large range. Moreover, the figure indicates that beyond a rating scale of 5, the performance increase slows significantly.

Table 6: Human Evaluation on the quality of criteria.

Dataset	Relevancy	Usefulness	Diversity
Trec-Covid	4.75	3.86	4.52
Touche	4.63	4.18	4.48
News	4.82	3.95	4.69
Average	4.73	4.00	4.56

5.5.3 Human Evaluation on the quality of criteria. To evaluate the quality of the criteria generated by MCRanker, we engaged two PhD students as human assessors. The evaluation process began with the random selection of 20 queries from the Trec-Covid, Touche, and News datasets. We then used MCRanker's Team Recruiting module to assemble a virtual annotation team, consisting of one NLP Scientist and two collaborators. The Criteria Generation module was then used to generate criteria for each team member. The human assessors were provided with both the queries and the corresponding criteria and were asked to evaluate three aspects: "Relevancy" (*relevance to the query*), "Usefulness" (*clarity and guidance provided*), and "Diversity" (*reflection of different domain expertise*) using a 0-5 rating scale. The results are presented in Table 6.

From this evaluation, we observe that the criteria generated by MCRanker received high scores for "Relevancy" and "Diversity", indicating that the virtual annotation team can produce distinct yet valuable criteria. However, the "Usefulness" score was somewhat lower, likely because MCRanker had access only to the query but not the associated passages. This limitation restricted MCRanker's ability to fully understand the meaning of the query, resulting in less appropriate criteria.

5.5.4 Sensitivity of the proposed method. To test the sensitivity of the proposed method, we ran MCRanker on the Trec-Covid, Touche, and News datasets ten times. We calculated the standard deviation of the results, which are 0.21 for Trec-covid, 0.15 for Touche, and

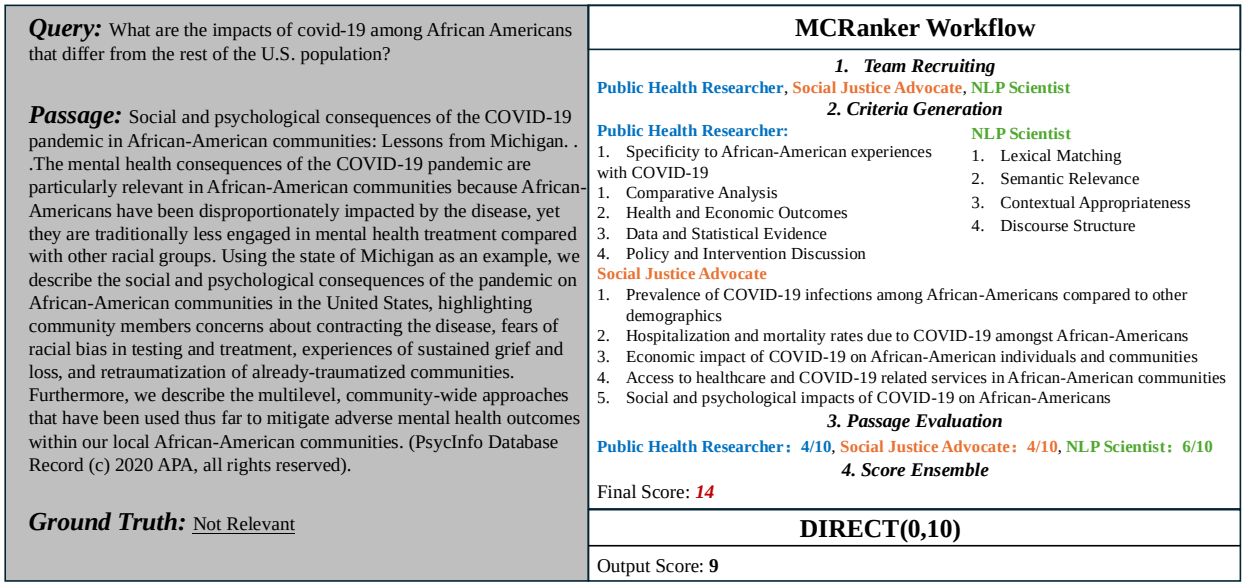

Figure 5: MCRanker spots a subtle semantic difference between the query and an irrelevant passage.

0.22 for News. The relatively low standard deviation values indicate that MCRanker maintains fairly consistent performance and is not sensitive to variations in different newly recruited R.C.s.

Table 7: R.C. Identities Recruited by MCRanker for the Query "How does the coronavirus respond to changes in the weather?".

No.	R.C. 1	R.C. 2
1	Public Health Researcher	Climate Change Advocate
2	Public Health Policy Maker	Environmental Researcher
3	Public Health Researcher	Meteorologist
4	Health Professional	Climate Change Researcher
5	Health Policy Analyst	Environmental Scientist

To further clarify MCRanker's sensitivity in recruiting R.C.s, we walk into a case to illustrate a typical behavior observed during the R.C. recruitment process. Using the query "How does the coronavirus respond to changes in the weather?" as an example, the specific identities generated are in Table 7. From these results, we can conclude that each time MCRanker recruited R.C.s, it consistently selected one R.C. related to climate (e.g., Climate Change Researcher, Meteorologist, Environmental Researcher) and another related to human health (e.g., Public Health Researcher, Health Policy Analyst). This demonstrates that, although there is some randomness in R.C. recruitment, the domains of the recruited R.C.s remain roughly consistent.

5.5.5 Case Study. Figure 5 illustrates how MCRanker effectively captures subtle semantic relationships between the query, "What are the impacts of COVID-19 among African Americans that differ from the rest of the U.S. population?" and an irrelevant passage. During the Team Recruiting phase, MCRanker built a virtual annotation team consisting of a fixed "NLP Scientist" and two recruited collaborators: a "Public Health Researcher" and a "Social Justice Advocate." After reading the query, each team member generated criteria that reflected their specific domain expertise, providing detailed guidance for the subsequent passage evaluation process. Based on these criteria, each member then gave a relevance score. In this case, both the "Public Health Researcher" and the "Social Justice Advocate" assigned a low relevance score, since in their criteria, both of them value comparative results in the passage and expect to see more detailed information like data evidence, economic impact, and social impact. The final score, obtained by summing the individual scores, was 14 out of 30. In contrast, the baseline DIRECT(0,10), which directly prompts an LLM to generate a relevance score for each query-passage pair, assigned a high score of 9 out of 10 to this irrelevant passage.

6 Conclusion

In this work, we explore how automatically generated query-based multi-perspective criteria can be used to overcome the inconsistent and biased prediction from zero-shot pointwise LLM rankers. Our experiments on BEIR benchmarks demonstrate our proposed method can consistently improve the ranking performance. This work is among the first to apply the concept of "multi-perspective problem solving" to a ranking task. Furthermore, our in-depth analysis reveals that quality criteria can robustly and significantly improve the performance of the pointwise ranker, even when built upon a less powerful base model.

For future work, we will investigate extending our findings to pairwise and listwise ranking frameworks and further explore the collaboration mechanisms underlying the virtual annotators.

References

[1] Mariam ALMutairi, Lulwah AlKulaib, Melike Aktas, Sara Alsalamah, and Chang-Tien Lu. 2024. Synthetic Arabic Medical Dialogues Using Advanced Multi-Agent LLM Techniques. In *Proceedings of The Second Arabic Natural Language Processing Conference*. 11–26.

[2] Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024. ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs. arXiv:2309.13007 [cs.CL]

[3] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan. 2023. Toxicity in chatgpt: Analyzing persona-assigned language models. *arXiv preprint arXiv:2304.05335* (2023).

[4] Guglielmo Faggioli, Laura Dietz, Charles Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2024. Who Determines What Is Relevant? Humans or AI? Why Not Both? A spectrum of human-AI collaboration in assessing relevance. *Commun. ACM* (2024).

[5] Guglielmo Faggioli, Laura Dietz, Charles LA Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on large language models for relevance judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*. 39–50.

[6] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as You Desire. arXiv:2302.04166 [cs.CL] https://arxiv.org/abs/2302.04166

[7] Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. 2023. Improving language model negotiation with self-play and in-context learning from ai feedback. *arXiv preprint arXiv:2305.10142* (2023).

[8] Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *Journal of Artificial Intelligence Research* 77 (2023), 103–166.

[9] Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. 2020. Learning-to-Rank with BERT in TF-Ranking. *arXiv preprint arXiv:2004.08476* (2020).

[10] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[11] Tae Soo Kim, Yoonjoo Lee, Jamin Shin, Young-Ho Kim, and Juho Kim. 2024. EvalLM: Interactive Evaluation of Large Language Model Prompts on User-Defined Criteria. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24, Vol. 35)*. ACM, 1–21. https://doi.org/10.1145/3613904.3642216

[12] Xiaochong Lan, Chen Gao, Depeng Jin, and Yong Li. 2023. Stance detection with collaborative role-infused llm-based agents. *arXiv preprint arXiv:2310.10467* (2023).

[13] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2023. PARADE: Passage Representation Aggregation forDocument Reranking. *ACM Transactions on Information Systems* 42, 2 (2023), 1–26.

[14] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760* (2023).

[15] Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2022. Branch-Train-Merge: Embarrassingly Parallel Training of Expert Language Models. arXiv:2208.03306 [cs.CL]

[16] Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, Yuxuan Lai, Chongyang Tao, and Shuai Ma. 2024. Leveraging Large Language Models for NLG Evaluation: Advances and Challenges. arXiv:2401.07103 [cs.CL] https://arxiv.org/abs/2401.07103

[17] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).

[18] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118* (2023).

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

[20] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. arXiv:2303.16634 [cs.CL] https://arxiv.org/abs/2303.16634

[21] Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023. Calibrating LLM-Based Evaluator. arXiv:2309.13308 [cs.CL] https://arxiv.org/abs/2309.13308

[22] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2024. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2421–2425.

[23] 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* (2023).

[24] Liam Magee, Vanicka Arora, Gus Gollings, and Norma Lam-Saw. 2024. The Drama Machine: Simulating Character Development with LLM Agents. arXiv:2408.01725 [cs.CY] https://arxiv.org/abs/2408.01725

[25] Varun Nair, Elliot Schumacher, Geoffrey Tso, and Anitha Kannan. 2023. DERA: Enhancing Large Language Model Completions with Dialog-Enabled Resolving Agents. arXiv:2303.17071 [cs.CL]

[26] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713* (2020).

[27] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[28] 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* (2023).

[29] Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. RankZephyr: Effective and Robust Zero-Shot Listwise Reranking is a Breeze! *arXiv preprint arXiv:2312.02724* (2023).

[30] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

[31] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2020. Are neural rankers still outperformed by gradient boosted decision trees?. In *International Conference on Learning Representations*.

[32] Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. arXiv:2402.14016 [cs.CL]

[33] 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* (2022).

[34] Yijia Shao, Yucheng Jiang, Theodore A Kanell, Peter Xu, Omar Khattab, and Monica S Lam. 2024. Assisting in Writing Wikipedia-like Articles From Scratch with Large Language Models. *arXiv preprint arXiv:2402.14207* (2024).

[35] Leixian Shen, Haotian Li, Yun Wang, and Huamin Qu. 2024. From Data to Story: Towards Automatic Animated Data Video Creation with LLM-based Multi-Agent Systems. *arXiv preprint arXiv:2408.03876* (2024).

[36] Malavikha Sudarshan, Sophie Shih, Estella Yee, Alina Yang, John Zou, Cathy Chen, Quan Zhou, Leon Chen, Chinmay Singhal, and George Shih. 2024. Agentic LLM Workflows for Generating Patient-Friendly Medical Reports. *arXiv preprint arXiv:2408.01112* (2024).

[37] 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* (2023).

[38] 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:2104.08663 [cs.IR] https://arxiv.org/abs/2104.08663

[39] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2023. Large language models can accurately predict searcher preferences. *arXiv preprint arXiv:2309.10621* (2023).

[40] Tina Tseng, Amanda Stent, and Domenic Maida. 2020. Best Practices for Managing Data Annotation Projects. *CoRR* abs/2009.11654 (2020). arXiv:2009.11654 https://arxiv.org/abs/2009.11654

[41] Chris Van Der Lee, Albert Gatt, Emiel Van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*. 355–368.

[42] Jianyou Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. 2023. DORIS-MAE: Scientific document retrieval using multi-level aspect-based queries. *arXiv preprint arXiv:2310.04678* (2023).

[43] Junda Wang, Zonghai Yao, Zhichao Yang, Huixue Zhou, Rumeng Li, Xun Wang, Yucheng Xu, and Hong Yu. 2024. NoteChat: A Dataset of Synthetic Patient-Physician Conversations Conditioned on Clinical Notes. In *Findings of the Association for Computational Linguistics ACL 2024*. 15183–15201.

[44] Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024. Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key? arXiv:2402.18272 [cs.CL]

[45] Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *arXiv preprint arXiv:2307.05300* (2023).

[46] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. [n.d.]. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. ([n. d.]).

[47] Ruicheng Xian, Honglei Zhuang, Zhen Qin, Hamed Zamani, Jing Lu, Ji Ma, Kai Hui, Han Zhao, Xuanhui Wang, and Michael Bendersky. 2023. Learning List-Level Domain-Invariant Representations for Ranking. *Advances in Neural Information Processing Systems* 36 (2023).

[48] Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688* (2023).

[49] Jintian Zhang, Xin Xu, and Shumin Deng. 2023. Exploring collaboration mechanisms for llm agents: A social psychology view. *arXiv preprint arXiv:2310.02124* (2023).

[50] Xinyu Zhang, Sebastian Hofstätter, Patrick Lewis, Raphael Tang, and Jimmy Lin. 2023. Rank-without-gpt: Building gpt-independent listwise rerankers on open-source large language models. *arXiv preprint arXiv:2312.02969* (2023).

[51] Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LongAgent: Scaling Language Models to 128k Context through Multi-Agent Collaboration. *arXiv preprint arXiv:2402.11550* (2024).

[52] Andrew Zhu, Liam Dugan, and Chris Callison-Burch. 2024. ReDel: A Toolkit for LLM-Powered Recursive Multi-Agent Systems. *arXiv preprint arXiv:2408.02248* (2024).

[53] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zhicheng Dou, and Ji-Rong Wen. 2024. Large Language Models for Information Retrieval: A Survey. arXiv:2308.07107 [cs.CL] https://arxiv.org/abs/2308.07107

[54] Honglei Zhuang, Zhen Qin, Shuguang Han, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2021. Ensemble distillation for BERT-based ranking models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 131–136.

[55] Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024. Beyond Yes and No: Improving Zero-Shot LLM Rankers via Scoring Fine-Grained Relevance Labels. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

[56] Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023. 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*. 2308–2313.

[57] 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*.