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

Fang Guo[1*], Wenyu Li [2*], Honglei Zhuang[3], Yun Luo[1], Yafu Li[1], Qi Zhu[4], Le Yan[3], and Yue Zhang[1]

[1]Department of Engineering, Westlake University
[2]Department of Future Technology, South China University of Technology
[3]Google Reserach
[4]Department of Computer Science, University of Illinois at Urbana-Champaign
[1]{*guofang,luoyun,liyafu,zhangyue*}@*westlake.edu.cn*
[2]{*wenyulilwy*}@*gmail.com*
[3]{*hlz,lyyanle*}@*google.com*
[4]{*qiz3*}@*illinois.edu*

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 complicated passages. To address these shortcomings, we propose to build a ranker that generates ranking scores based on a set of criteria from various perspectives. These criteria are intended to direct each perspective in providing a distinct yet synergistic evaluation. Our research, which examines eight datasets from the BEIR benchmark demonstrates that incorporating this multi-perspective criteria ensemble approach markedly enhanced the performance of pointwise LLM rankers[1].

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 (Qin et al., 2024; Sun et al., 2023; Zhuang et al., 2023; Pradeep et al., 2023a; Li et al., 2023a). Nevertheless, the iterative nature of querying LLMs, combined with their stochastic behavior, leads to inconsistent assessment criteria for zero-shot rankers, resulting in a lack of both *consistency* and *comprehensiveness* in the ranking process.

In the realm of LLM rankers, a pointwise ranker, which evaluates each passage individually without comparing its position or relation to other passages, owns its advantage in scalability and interpretability (Liang et al., 2022; Zhuang et al., 2024a; Sachan et al., 2022). However, the pointwise ranker is not immune to the issues of inconsistent and biased assessments of passages. 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 topic erroneously receives a high score. This error happens when the LLM ranker decides to adopt a biased assessment criterion that prioritizes keyword presence over the nuanced understanding of content semantics. In this paper, we investigate

[*] These two authors contribute equally.
[1]Code, data, and prompts can be found at: https://github.com/fangguo1/MCRanker

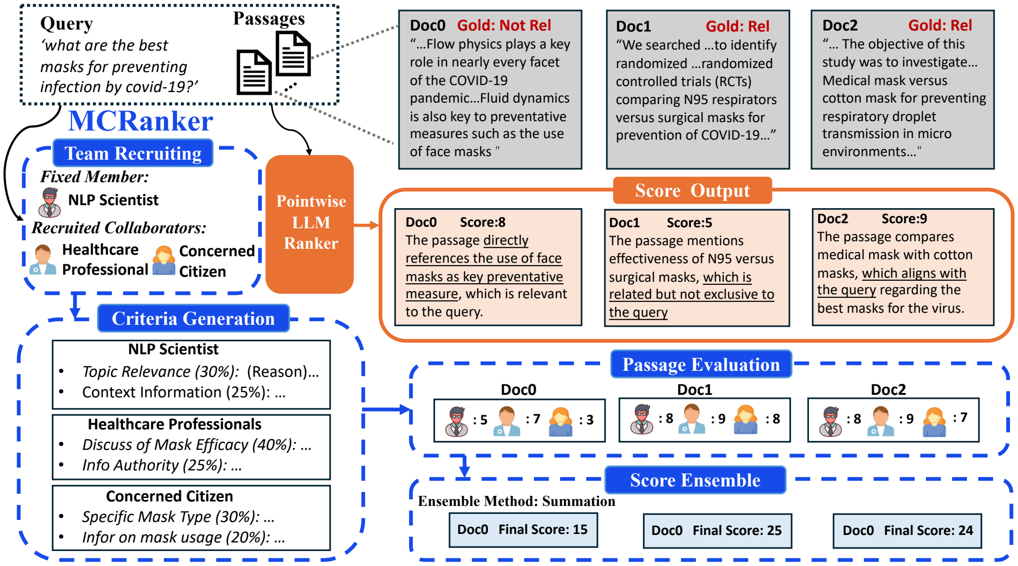

Query
'what are the best masks for preventing infection by covid-19?'
Passages
MCRanker
Team Recruiting
Fixed Member:
NLP Scientist
Recruited Collaborators:
Healthcare Professional
Concerned Citizen
Pointwise LLM Ranker
Doc0 Gold: Not Rel
"...Flow physics plays a key role in nearly every facet of the COVID-19 pandemic...Fluid dynamics is also key to preventative measures such as the use of face masks "
Doc1 Gold: Rel
"We searched ...to identify randomized ...randomized controlled trials (RCTs) comparing N95 respirators versus surgical masks for prevention of COVID-19..."
Doc2 Gold: Rel
"... The objective of this study was to investigate... Medical mask versus cotton mask for preventing respiratory droplet transmission in micro environments..."
Score Output
Doc0 Score:8
The passage directly references the use of face masks as key preventative measure, which is relevant to the query.
Doc1 Score:5
The passage mentions effectiveness of N95 versus surgical masks, which is related but not exclusive to the query
Doc2 Score:9
The passage compares medical mask with cotton masks, which aligns with the query regarding the best masks for the virus.
Criteria Generation
NLP Scientist
Topic Relevance (30%): (Reason)...
Context Information (25%): ...
Healthcare Professionals
Discuss of Mask Efficacy (40%): ...
Info Authority (25%): ...
Concerned Citizen
Specific Mask Type (30%): ...
Infor on mask usage (20%): ...
Passage Evaluation
Doc0
:5 :7 :3
Doc1
:8 :9 :8
Doc2
:8 :9 :7
Score Ensemble
Ensemble Method: Summation
Doc0 Final Score: 15
Doc0 Final Score: 25
Doc0 Final Score: 24

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.

how to build a zero-shot pointwise ranker that is capable of generating both consistent and comprehensive assessments of passages.

Recent studies on human evaluative practices (Tseng et al., 2020; Gehrmann et al., 2023) suggest that optimal annotation outcomes are achieved through the collaboration of annotators with diverse expertise, bolstered by a standardized annotation guideline. On one hand, optimal annotations necessitate a synergy between domain-specific experts and language experts to guarantee the integrity of the evaluation. For example, when annotating the TREC COVID dataset (Voorhees et al., 2020), the annotation team included both medical students and professional indexers. On the other hand, the absence or ambiguity of annotation criteria could significantly diminish the quality and efficiency of the annotation process (Van Der Lee et al., 2019; Gehrmann et al., 2023). Consequently, it is imperative to establish precise, well-articulated criteria before the commencement of the annotation process.

In line with these guidelines for humans, we introduce the MCRanker framework, which generates **M**ulti-perspective **C**riteria to improve pointwise LLM **Ranker**. 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 emulates the domain expertise and text analytical capabilities of professional annotators through virtual team recruiting. 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 ensembled through aggregation, yielding a 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, the first model to integrate a multi-perspective ideology into a ranking task. It simulates the professional human annotation process by building an annotation team that has both text analysis skills and diverse domain expertise.

- We investigate how criteria can be generated and utilized to influence a ranker. The in-depth analysis further shows 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 research has been investigating the application of large language models (LLMs) for zero-shot text ranking (Nogueira et al., 2019; Han et al., 2020; Zhuang et al., 2021; Nogueira et al., 2020a; Zhuang et al., 2023; Xian et al., 2023; Liu et al., 2009; Qin et al., 2020; Zhang et al., 2023b). Notably, pairwise (Qin et al., 2024) and listwise (Sun et al., 2023; Ma et al., 2023; Zhuang et al., 2024b; Pradeep et al., 2023a;b) 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 (Liang et al., 2022) or the probability of generating the query from the document (Sachan et al., 2022). 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.

Zero-shot LLM Assessors Recent studies have explored the LLMs' capability as pseudo-assessors (Faggioli et al., 2023; 2024; Raina et al., 2024; Wang et al., 2023a; Thomas et al., 2023). 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 (Deshpande et al., 2023; Xu et al., 2023; Fu et al., 2023; Li et al., 2023b; Wang et al., 2024; Nair et al., 2023) and "Mixture-of-Experts" systems (Li et al., 2022), 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 (Wang et al., 2023b; Shao et al., 2024) or manual selection (Lan et al., 2023; Zhao et al., 2024; Zhang et al., 2023a). These identified LLM agents follow various collaboration mechanisms (Chen et al., 2024) to accomplish given tasks.

Inspired by this "Multi-Perspective" philosophy, we design a "Team Recruiting" step, which automatically generates multiple collaborators to work with a fixed NLP scientist. In this work, we did not study the collaboration mechanism within the team and leave it for future work.

3 Methods

In this section, we first revisit pointwise LLM rankers. Then we introduce our proposed MCRanker in detail. The pipeline of MCRanker is shown in Figure 1.

3.1 Preliminary

We formally describe how a 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 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 Method

To make a pointwise ranker evaluating query-passage pairs in consistent and comprehensive criteria, we propose MCRanker which instructs LLM to perform the following procedures: (1) **Team Recruiting** that generates 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 ensembled from team members' evaluation results. Figure 1 shows a working example of MCRanker during inference.

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 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. It can be found in Appendix D.1.

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 $M_{Criteria}$ and can be found in Appendix D.2 and D.3.

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 and can be found in Appendix D.4.

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) Rank Ensemble, 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) Rank Ensemble, 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:

$$s_{\text{Sum},i} = \sum_{j=0}^{|A|} s_{ij} \quad s_{\text{RE},i} = \sum_{j=0}^{|A|} \frac{1}{r_i^j} \quad s_{\text{Assessor},i} = M_{Assessor}(x_a(q, p_i, s_{i0}, \ldots, s_{ij}, r_{i0}, \ldots, r_{ij})) \tag{4}$$

where x_a is the prompt for the final score assessment and is shown in Appendix D.5. The final output of MCRanker is a ranked list on s_i in descending order.

4 Experimental Setup

4.1 Dataset

Same as (Zhuang et al., 2024a), our experiments were conducted on 8 datasets from BEIR benchmark: 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) (Sachan et al., 2022)**: 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. **Binary Relevance Generation (RG-YN) (Liang et al., 2022)**: It prompts the LLM to predict on a query-passage pair and uses the likelihood of "Yes/No" for ranking score calculation.

3. **Rating Scale 0-to-k Relevance Generation (RG-S) (Zhuang et al., 2024a)**: 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. The prompt we use is shown in Appendix D.6.

Notice that QG, RG-YN, and RG-S are all likelihood-based pointwise rankers, and DIRECT(0, k) is text-based.

Table 1: Overall ranking performances measured by average NDCG@10 on BEIR data sets. The best performances are bolded, and the second are underlined. NLP Sci. represents NLP Scientist, R.C. represents the Recruited Collaborator, and a suffix *MC* represents our MCRanker with different amount of team members. * means likelihood-based models that utilize the token probability for relevance score calculation.

Method	Covid	Touche	News	Signal	DBPedia	SciFact	Robust04	NFCorpus	Avg
QG*	73.57	24.08	41.56	28.72	37.73	74.95	46.51	36.73	45.48
RG-YN*	78.97	24.27	45.88	31.96	36.96	69.58	56.56	37.43	47.70
RG-S*	80.48	27.57	47.90	**33.01**	41.90	**75.21**	56.68	39.01	50.22
DIRECT(0, 10)	79.30	25.22	46.19	29.12	40.82	70.08	53.78	37.52	47.75
DIRECT(0, 20)	79.96	22.05	47.57	27.77	40.53	70.53	54.66	37.19	47.53
(NLP Sci.)$_{MC}$	81.49	29.33	46.13	29.48	41.25	70.86	56.37	38.25	49.14
(1 R.C.)$_{MC}$	82.43	30.60	48.52	26.58	41.11	71.35	55.78	37.93	49.28
(2 R.C.)$_{MC}$	83.53	31.42	**50.90**	26.85	42.33	71.86	56.84	38.36	50.26
(NLP Sci. + 1 R.C.)$_{MC}$	84.16	**32.91**	49.54	29.94	43.85	73.33	57.12	39.12	51.24
(NLP Sci. + 2 R.C.)$_{MC}$	**84.23**	32.48	50.32	29.73	**44.67**	73.14	**57.23**	**39.58**	**51.42**

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 (Järvelin & Kekäläinen, 2002).

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 QG, RG-YN, and RG-S, we copied their performance on BEIR directly from the original papers. For $M_{Evaluation}$, we prompt it to output a score without further explaining the reason (Appendix D.4), this setup is in line with RG-S.

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 got the best performance and outperformed 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, which incorporates a virtual annotation team comprising an NLP scientist and one or two additional collaborators, 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 variant of MCRanker that utilizes additional two collaborators besides an NLP scientist (NLP Sci. + 2 R.C.)$_{MC}$ shows an increase in performance over the single-collaborator one (NLP Sci. + 1 R.C.)$_{MC}$ in five datasets, although this improvement is not statistically substantial. This may be attributed

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

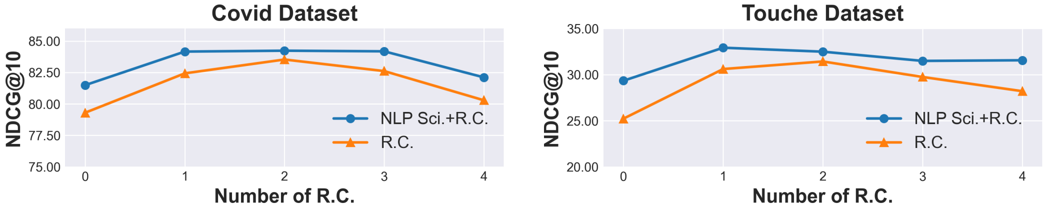


Figure 2: Research on different number of Team Member

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 synergistic 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 hypothesize that if we could get the token probability from the LLM, the performance of MCRanker can be further enhanced.

5.2 Ablation Study

To investigate the impact of each module in our design, we conducted ablation studies to assess the performance of our framework when each module was removed or switched to a different configuration. In the following, we provide a detailed description of the results.

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.

The results indicate that the inclusion of any team member like only one NLP scientist variant (NLP Sci.)$_{MC}$ or only one recruited collaborator (1 R.C.)$_{MC}$ 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 (2 R.C.)$_{MC}$ consistently overperform (1 R.C.)$_{MC}$ 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 (NLP Sci.)$_{MC}$ achieves performance comparable to the single recruited collaborator variant (1 R.C.)$_{MC}$ on average, a notable performance gain is observed when the (NLP Sci. + 1 R.C.)$_{MC}$ model is compared to the (2 R.C.)$_{MC}$ variant. The former shows a greater synergistic effect. 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 might be due to redundancy in expertise among an increased number of collaborators. We leave the investigation of the underlying cause for future study.

Table 2: Ablation study on criteria utility measured by average NDCG@10.

Method	**Covid**	**Touche**
DIRECT(0, 10)	79.30	25.22
DIRECT(0, 10)+criteria	79.86	30.46
MCRanker without criteria	79.97	26.22
MCRanker	**84.23**	**32.48**

Table 3: Ablation study on broad creteria measured by average NDCG@10. "DBC" represent "Dataset-Based-Criteria".

Method	**Covid**	**Touche**
DIRECT(0, 10)	79.30	25.22
MCRanker-DBC	82.28	27.78
MCRanker	**84.23**	**32.48**

Table 4: Ablation study on ensemble mechanism measured by average NDCG@10.

Method	**Covid**	**Touche**
MCRanker-RE	82.55	30.73
MCRanker-$M_{Assessor}$	83.58	31.35
MCRanker	**84.23**	**32.48**

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 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. Conversely, stripping MCRanker of its criteria results in a marked performance decline, with decreases of 4.26 and 6.26 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 that can easily lead LLM to adopt inconsistent scoring guidelines.

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 $M_{Criteria}$. 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 and 4.70 for Touche. 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, MCRanker-DBC still records a notable performance improvement in comparison to DIRECT(0,10), demonstrating the importance of establishing criteria before scoring. This exploration can shed light on how to balance budget, specificity, and generality in criteria generation.

5.3 Further Analysis

In this section, we analyze the impact of different ensemble mechanisms, model performance with different rating scales, and the model's generalizability to different LLMs. Due to the space limit, we put two concrete examples of MCRanker's workflow in the Appendix Figure 5 and Figure 6, and a visualization of the identities of the generated collaborators in the Appendix Figure 7.

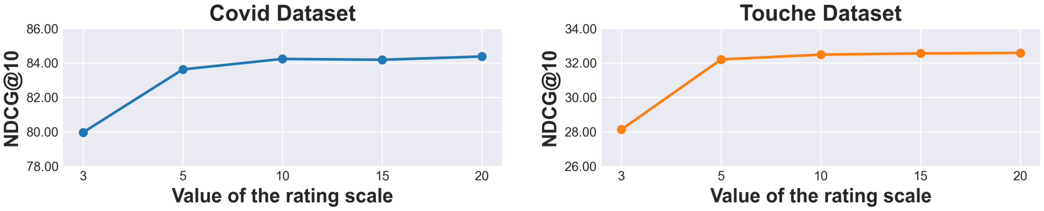


Figure 3: MCRanker with different values of the rating scale k

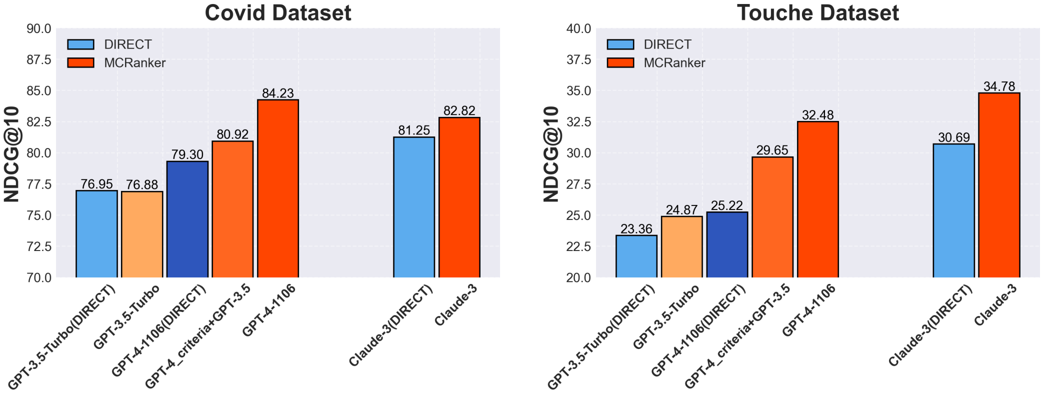


Figure 4: Comparing performance of different base models

5.3.1 Study on ensemble mechanism

We examine the effect of various ensemble strategies on the performance of MCRanker.The results, detailed in Table 4, indicate that the straightforward ”Score Summation” approach surpasses alternative methods. The limited effectiveness of the ”Ranking Ensemble” 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.

5.3.2 Study on the variance of rating scale

We plot how the performance changes about the rating scale for our proposed MCRanker in Figure 3. 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.

5.3.3 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 Table 4. 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. It underscores the robust generalizability of our proposed multi-perspective criteria ensemble methodology.

6 Conclusion

In this work, we explore the use of a multi-perspective criteria ensemble method to overcome the inconsistent and biased prediction from 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 LLM ranker, even when built upon a less powerful base model.

For future work, we will investigate how to extend our discovery to other ranking frameworks and further explore the collaboration mechanism behind the virtual annotators.

References

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms, 2024.

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

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

Guglielmo Faggioli, Laura Dietz, Charles Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. Who determines what is relevant? humans or ai? why not both? a spectrum of human-ai collaboration in assessing relevance. *Communications of the ACM*, 2024.

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

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

Shuguang Han, Xuanhui Wang, Mike Bendersky, and Marc Najork. Learning-to-rank with bert in tf-ranking. *arXiv preprint arXiv:2004.08476*, 2020.

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

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

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. Parade: Passage representation aggregation fordocument reranking. *ACM Transactions on Information Systems*, 42(2):1–26, 2023a.

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

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. Branch-train-merge: Embarrassingly parallel training of expert language models, 2022.

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

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

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

Varun Nair, Elliot Schumacher, Geoffrey Tso, and Anitha Kannan. Dera: Enhancing large language model completions with dialog-enabled resolving agents, 2023.

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

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

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model, 2020b.

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

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze! *arXiv preprint arXiv:2312.02724*, 2023b.

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

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. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024.

Vyas Raina, Adian Liusie, and Mark Gales. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment, 2024.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. Improving passage retrieval with zero-shot question generation. *arXiv preprint arXiv:2204.07496*, 2022.

Yijia Shao, Yucheng Jiang, Theodore A Kanell, Peter Xu, Omar Khattab, and Monica S Lam. Assisting in writing wikipedia-like articles from scratch with large language models. *arXiv preprint arXiv:2402.14207*, 2024.

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

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

Tina Tseng, Amanda Stent, and Domenic Maida. Best practices for managing data annotation projects. *CoRR*, abs/2009.11654, 2020. URL https://arxiv.org/abs/2009.11654.

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

Ellen M. Voorhees, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, William R. Hersh, Kyle Lo, Kirk Roberts, Ian Soboroff, and Lucy Lu Wang. TREC-COVID: constructing a pandemic information retrieval test collection. *CoRR*, abs/2005.04474, 2020. URL https://arxiv.org/abs/2005.04474.

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

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key?, 2024.

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

Ruicheng Xian, Honglei Zhuang, Zhen Qin, Hamed Zamani, Jing Lu, Ji Ma, Kai Hui, Han Zhao, Xuanhui Wang, and Michael Bendersky. Learning list-level domain-invariant representations for ranking. *Advances in Neural Information Processing Systems*, 36, 2023.

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

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

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

Jun Zhao, Can Zu, Hao Xu, Yi Lu, Wei He, Yiwen Ding, Tao Gui, Qi Zhang, and Xuanjing Huang. Longagent: Scaling language models to 128k context through multi-agent collaboration. *arXiv preprint arXiv:2402.11550*, 2024.

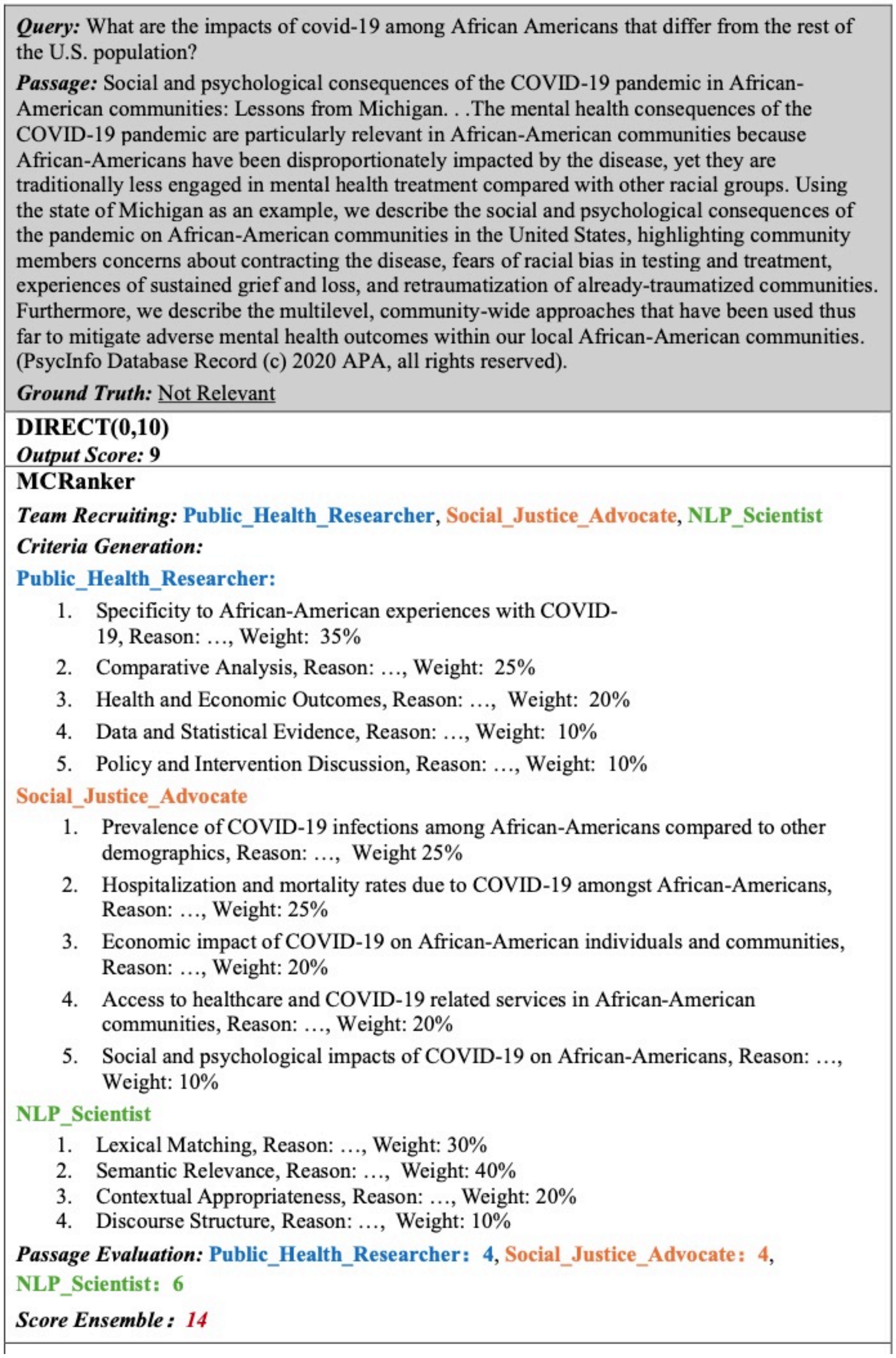
Honglei Zhuang, Zhen Qin, Shuguang Han, Xuanhui Wang, Michael Bendersky, and Marc Najork. Ensemble distillation for bert-based ranking models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 131–136, 2021.

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

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 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*, 2024a.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 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*, 2024b.

A Concrete Example of MCRanker



Query: What are the impacts of covid-19 among African Americans that differ from the rest of the U.S. population?

Passage: Social and psychological consequences of the COVID-19 pandemic in African-American communities: Lessons from Michigan. . .The mental health consequences of the COVID-19 pandemic are particularly relevant in African-American communities because African-Americans have been disproportionately impacted by the disease, yet they are traditionally less engaged in mental health treatment compared with other racial groups. Using the state of Michigan as an example, we describe the social and psychological consequences of the pandemic on African-American communities in the United States, highlighting community members concerns about contracting the disease, fears of racial bias in testing and treatment, experiences of sustained grief and loss, and retraumatization of already-traumatized communities. Furthermore, we describe the multilevel, community-wide approaches that have been used thus far to mitigate adverse mental health outcomes within our local African-American communities. (PsycInfo Database Record (c) 2020 APA, all rights reserved).

Ground Truth: Not Relevant

DIRECT(0,10)

***Output Score:* 9**

MCRanker

***Team Recruiting:* Public_Health_Researcher, Social_Justice_Advocate, NLP_Scientist**

Criteria Generation:

Public_Health_Researcher:

1. Specificity to African-American experiences with COVID-19, Reason: …, Weight: 35%
2. Comparative Analysis, Reason: …, Weight: 25%
3. Health and Economic Outcomes, Reason: …, Weight: 20%
4. Data and Statistical Evidence, Reason: …, Weight: 10%
5. Policy and Intervention Discussion, Reason: …, Weight: 10%

Social_Justice_Advocate

1. Prevalence of COVID-19 infections among African-Americans compared to other demographics, Reason: …, Weight 25%
2. Hospitalization and mortality rates due to COVID-19 amongst African-Americans, Reason: …, Weight: 25%
3. Economic impact of COVID-19 on African-American individuals and communities, Reason: …, Weight: 20%
4. Access to healthcare and COVID-19 related services in African-American communities, Reason: …, Weight: 20%
5. Social and psychological impacts of COVID-19 on African-Americans, Reason: …, Weight: 10%

NLP_Scientist

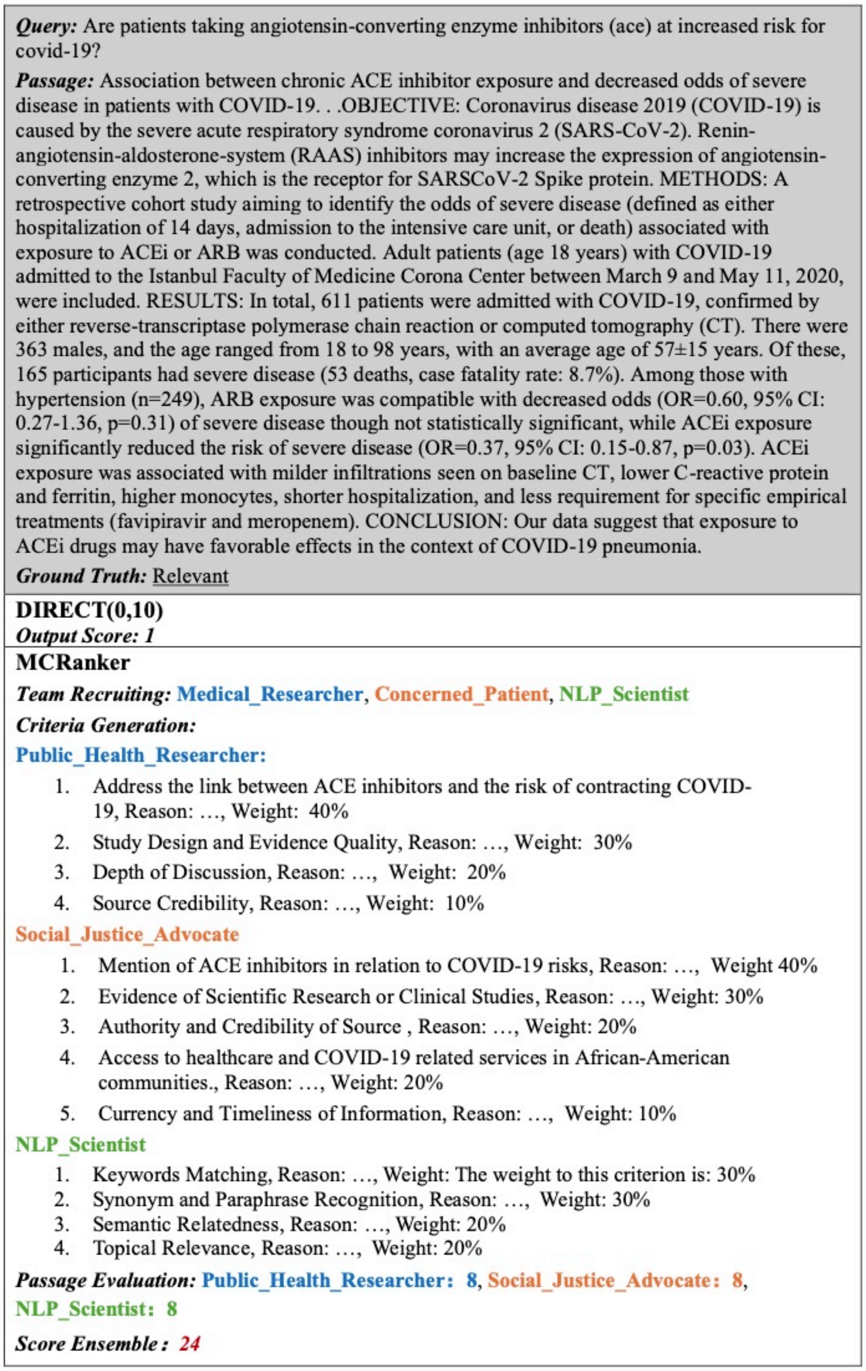
1. Lexical Matching, Reason: …, Weight: 30%
2. Semantic Relevance, Reason: …, Weight: 40%
3. Contextual Appropriateness, Reason: …, Weight: 20%
4. Discourse Structure, Reason: …, Weight: 10%

***Passage Evaluation:* Public_Health_Researcher： 4, Social_Justice_Advocate： 4, NLP_Scientist： 6**

Score Ensemble : ***14***

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

We present two specific query-passage pairs from the COVID dataset to showcase the functionality of MCRanker's counterparts and highlight how they yield a score that diverges from that of the DIRECT baseline. The first case is shown in Figure 5. The query is about the comparison between the African-American community and the rest of the U.S. population during COVID-19 and an irrelevant passage is mainly about the consequences of COVID-19



Query: Are patients taking angiotensin-converting enzyme inhibitors (ace) at increased risk for covid-19?

Passage: Association between chronic ACE inhibitor exposure and decreased odds of severe disease in patients with COVID-19. . .OBJECTIVE: Coronavirus disease 2019 (COVID-19) is caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Renin-angiotensin-aldosterone-system (RAAS) inhibitors may increase the expression of angiotensin-converting enzyme 2, which is the receptor for SARSCoV-2 Spike protein. METHODS: A retrospective cohort study aiming to identify the odds of severe disease (defined as either hospitalization of 14 days, admission to the intensive care unit, or death) associated with exposure to ACEi or ARB was conducted. Adult patients (age 18 years) with COVID-19 admitted to the Istanbul Faculty of Medicine Corona Center between March 9 and May 11, 2020, were included. RESULTS: In total, 611 patients were admitted with COVID-19, confirmed by either reverse-transcriptase polymerase chain reaction or computed tomography (CT). There were 363 males, and the age ranged from 18 to 98 years, with an average age of 57±15 years. Of these, 165 participants had severe disease (53 deaths, case fatality rate: 8.7%). Among those with hypertension (n=249), ARB exposure was compatible with decreased odds (OR=0.60, 95% CI: 0.27-1.36, p=0.31) of severe disease though not statistically significant, while ACEi exposure significantly reduced the risk of severe disease (OR=0.37, 95% CI: 0.15-0.87, p=0.03). ACEi exposure was associated with milder infiltrations seen on baseline CT, lower C-reactive protein and ferritin, higher monocytes, shorter hospitalization, and less requirement for specific empirical treatments (favipiravir and meropenem). CONCLUSION: Our data suggest that exposure to ACEi drugs may have favorable effects in the context of COVID-19 pneumonia.

Ground Truth: Relevant

DIRECT(0,10)

Output Score: 1

MCRanker

Team Recruiting: **Medical_Researcher**, **Concerned_Patient**, **NLP_Scientist**

Criteria Generation:

Public_Health_Researcher:

1. Address the link between ACE inhibitors and the risk of contracting COVID-19, Reason: …, Weight: 40%
2. Study Design and Evidence Quality, Reason: …, Weight: 30%
3. Depth of Discussion, Reason: …, Weight: 20%
4. Source Credibility, Reason: …, Weight: 10%

Social_Justice_Advocate

1. Mention of ACE inhibitors in relation to COVID-19 risks, Reason: …, Weight 40%
2. Evidence of Scientific Research or Clinical Studies, Reason: …, Weight: 30%
3. Authority and Credibility of Source , Reason: …, Weight: 20%
4. Access to healthcare and COVID-19 related services in African-American communities., Reason: …, Weight: 20%
5. Currency and Timeliness of Information, Reason: …, Weight: 10%

NLP_Scientist

1. Keywords Matching, Reason: …, Weight: The weight to this criterion is: 30%
2. Synonym and Paraphrase Recognition, Reason: …, Weight: 30%
3. Semantic Relatedness, Reason: …, Weight: 20%
4. Topical Relevance, Reason: …, Weight: 20%

Passage Evaluation: **Public_Health_Researcher: 8**, **Social_Justice_Advocate: 8**, **NLP_Scientist: 8**

Score Ensemble : 24

Figure 6: MCRanker's virtual annotation team conducts a more nuanced analysis of a relevant passage

in the African-American community. This subtle difference is spotted by MCRanker when most team members' criteria emphasize the "comparison" -related criterion. The second case, illustrated in Figure 6, involves a scenario where the DIRECT baseline incorrectly assigns a false negative prediction because the passage does not provide a direct response to the query. In contrast, MCRanker's virtual annotation team conducts a more nuanced analysis of the passage, leading to a high relevance score output.

B More Comparison Results

We also include a more thorough comparison with other methods including:

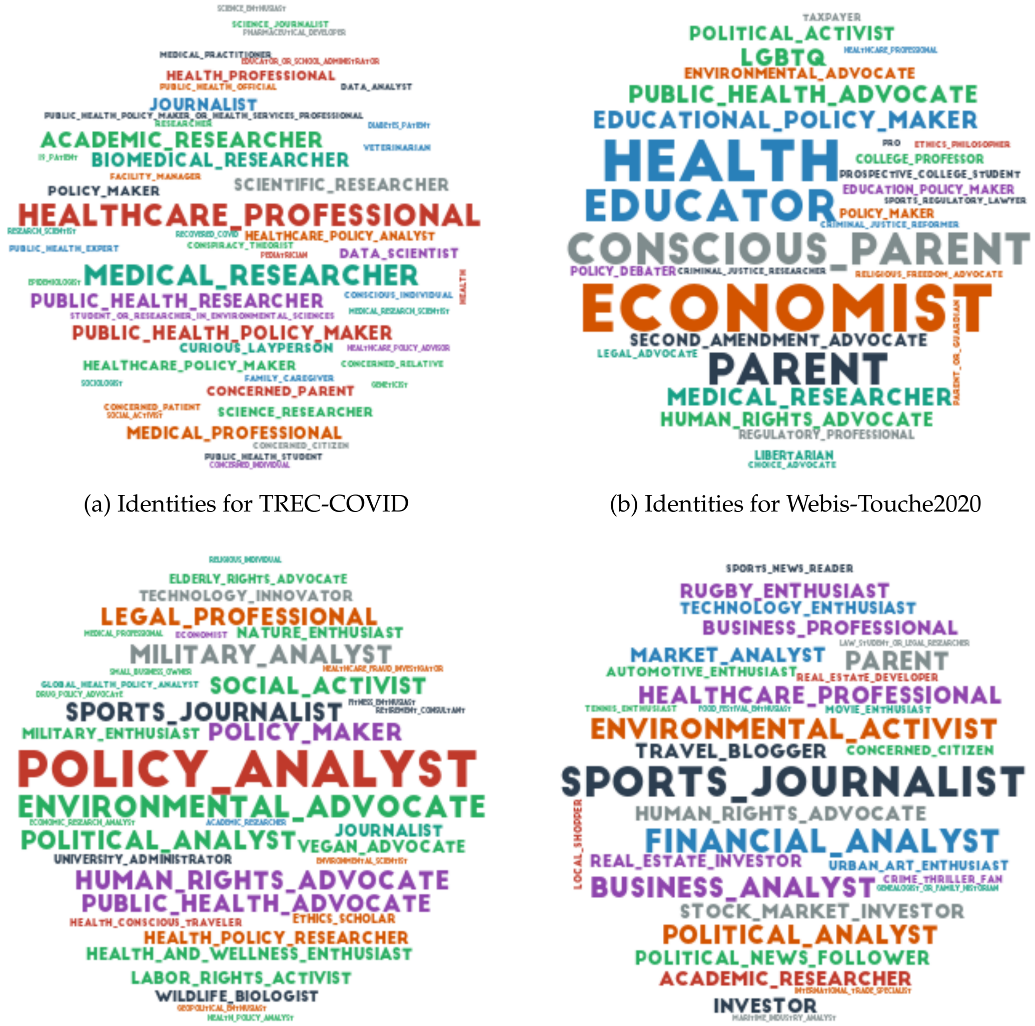
1. **BM25**: The base retriever performance.
2. **monoT5** (Nogueira et al., 2020b): A model based on T5 XL architecture and fine-tuned on the MS MARCO dataset. We test the model directly without further fine-tuning.
3. **RankT5** (Zhuang et al., 2023): This method uses an encoder-only model and employs a listwise softmax cross-entropy loss for ranking. It is fine-tuned on the MS MARCO dataset. We test the model directly without further fine-tuning.
4. **RankGPT** (Sun et al., 2023): A zero-shot listwise LLM ranker that processes a query and a set of documents, producing a ranking of the documents based on their relevance. This method incorporates a sliding window approach. Besides the model that uses "GPT-3.5-Turbo" as the base LLM, we further include a "GPT-4" variant of it that does a second-stage ranking based on the "GPT-3.5-Turbo" version's ranking result.

Table 5: Overall ranking performances measured by NDCG@10 on BEIR.

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
RG-YN	FLAN PaLM2 S	78.97	24.27	45.88	31.96	36.96	69.58	56.56	37.43	47.70
RG-S(0,2)	FLAN PaLM2 S	77.60	26.95	46.77	30.34	37.09	69.21	55.57	37.87	47.68
RG-S(0,4)	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
(NLP Sci.)$_{MC}$	GPT-4-1106-Preview	81.49	29.33	46.13	29.48	41.25	70.86	56.37	38.25	49.14
(1 R.C.)$_{MC}$	GPT-4-1106-Preview	82.43	30.60	48.52	26.58	41.11	71.35	55.78	37.93	49.28
(2 R.C.)$_{MC}$	GPT-4-1106-Preview	83.53	31.42	50.90	26.85	42.33	71.86	56.84	38.36	50.26
(NLP Sci. + 1 R.C.)$_{MC}$	GPT-4-1106-Preview	84.16	32.91	49.54	29.94	43.85	73.33	57.12	39.12	51.24
(NLP Sci. + 2 R.C.)$_{MC}$	GPT-4-1106-Preview	84.23	32.48	50.32	29.73	44.67	73.14	57.23	39.58	51.42
monoT5	Fine-tuned T5 XL	80.71	32.41	48.49	32.55	44.45	76.57	56.71	38.97	51.36
RankT5	Fine-tuned T5 XL	82.00	37.62	48.15	31.80	44.19	76.86	52.76	38.60	51.50
RankGPT	- GPT-3.5-Turbo	76.67	36.18	48.85	32.12	44.47	70.43	50.62	35.62	49.37
	- GPT-4	85.51	38.47	52.89	34.40	47.12	74.95	57.55	38.47	53.68

From Table 5, we can observe that MCRanker achieves compatible performance with fine-tuned rankers like monoT5 and RankT5. Also, it is encouraging to see that MCRanker outperforms listwise RankGPT with a "GPT-3.5-Turbo" base model and substantially shrinks the gap between the "GPT-4" version RankGPT.

C Team Member Identity Visualization



(a) Identities for TREC-COVID

(b) Identities for Webis-Touche2020

(c) Identities for News

(d) Identities for Signal

Figure 7: Visualization of the identities of Recruited Collaborator (R.C.) on four datasets. The identities show a high correlation with the nature of the datasets.

4

D Prompts

D.1 Team Recruiting

The task is when given a query and a passage example, to try to guess the most probable user identities. You can name up to number identities. Try to make these identities very different from each other. Note that the passage example is just to show you the passage style and the user could possibly like/dislike it. So do not let the stance in the passage influence your guess.

Given query:<<<{query}>>>

Given passage:<<<{passage}>>>

Please give your output in JSON format with keys are \“Identities\” and \“Reason\” .

Under the content of \“Identities\”, please output the identity, your identity should be correct, clear,and easy to understand

Under the content of \“Reason\”, explain why you output these identities.

Provide a clear and concise response, just give your answer in JSON format as I request, and don’t say any other words.

D.2 NLP Scientist Criteria Generation

As an NLP Scientist, you are good at judging the linguistic relevance of a passage in relation to the given query, listing the criteria for judging linguistic relevance between the query and a passage, and assigning weights to each criterion.

Given query:<<<{query}>>>

Please give your output in JSON format with keys are \“Criteria\” and \“Reason\” .

Under the content of \“Criteria\”, please output the criteria and some explanation to it, your criteria should be correct, clear, executable, and easy to understand, the criteria should come from your knowledge as the the NLP Scientist. The weight to every criterion should be added after each criterion with ’The weight to this criterion is:’.

Under the content of \“Reason\”, explain why you output these criteria.

Focus solely on your NLP Scientist expertise and avoid deductions outside of your expert criteria.

Provide a clear and concise response, just give your answer in JSON format as I request, and don’t say any other words.

D.3 Recruited Collaborators Criteria Generation

As a <<<{identity}>>>, you are asked to judge the relevance of a passage in relation to the given query, list the criteria for judging relevance between the query and a passage, assign weights to each criterion.

Given query: <<<{query}>>>

Please give your output in JSON format with keys \“Criteria\” and \“Reason\”. Under the content of \“Criteria\”, please output the criteria and some explanation to it,

your criteria should be correct, clear, executable, and easy to understand, the criteria should come from your knowledge as the <<<{identity}>>>. The weight to every criterion should be added after each criterion with ’The weight to this criterion is:’.

Under the content of \“Reason\”, explain why you output these criteria.

Focus solely on your identity and avoid deductions outside of your <<<{identity}>>>.

Provide a clear and concise response, just give your answer in JSON format as I request, don’t say any other words.

D.4 Team Member Score

Ignore all previous instructions.

Role: You are a <<<{identity}>>>

Task Description: I will give you the criteria, one passage, and a query, you should follow the criteria to analyze the given passage’s relevance from the view as a <<<{identity}>>>. Generate your answer in JSON format, with the key is \“Score\”. Under the content of \“Score\”, give me an integer score from 0 to 10 to represent the <<<{identity}>>>’s view relevance degree, just give the score, don’t say any other words.

Input-Output Description: The input will be the criteria, a passage, and a query, generate your output as I request:

Your Answer:

query: <<<{query}>>>

criteria: <<<{criteria}>>>

passage: <<<{passage}>>>

Output:

D.5 LLM Assessor

Ignore all previous instructions.

Role: You are adept at synthesizing diverse viewpoints to reach a well-considered conclusion.

Task Desciption: Your task is to evaluate the relevance of a given passage to a specified query. You will receive a passage, a query, and some relevance assessments from experts. These experts come from different fields and may not always agree. After reviewing their assessments, you are to integrate their insights and determine a final relevance score. Please give your output in JSON format,with the key is \“Final score\”. Under the content of \“Final score\” after carefully thinking about the two experts’ scores, combine them and just give one final relevance score, the score should be an integer from 0 to 10, don’t say any other words. Give your output in JSON as I request, don’t explain,don’t say any other words.

Input-Output Description: The input will be a passage, a query and three relevance assessments from three experts, generate your output as I request:

Your Answer:

query: <<<{query}>>>

passage: <<<{passage}>>>

Assessment of the three experts: Assessments from expert 1 who is an expert in <<<{identity}>>>: Relevance Score 1 (From 0 to 10): <<<{score}>>>

Assessments from expert 2 who is an expert in <<<{identity}>>> : Relevance Score 2 (From 0 to 10): <<<{score}>>>

Assessments from expert 3 who is an expert in language: Relevance Score 3 (From 0 to 10): <<<{score}>>>

Output:

D.6 Rating Scale 0-to-k Directly Score (DIRECT(0, k))

Ignore all previous instructions.

Role: You are an expert in judging relevance.

Task Description: From a scale of 0 to 10, judge the relevance between the query and the document. Give your output in JSON format, with the key is \“Score\”. Under the content of \“Score\”, just give me one final integer score from 0 to 10, don’t say any other words.

Input-Output Description: The input will be a query and a document.

Your Answer:

query: <<<{query}>>>

passage: <<<{passage}>>>

Output: