JudgeBlender: Ensembling Automatic Relevance Judgments

Hossein A. Rahmani
University College London
London, UK
hossein.rahmani.22@ucl.ac.uk

Emine Yilmaz
University College London
London, UK
emine.yilmaz@ucl.ac.uk

Nick Craswell
Microsoft
Seattle, US
nickcr@microsoft.com

Bhaskar Mitra
Microsoft
Montréal, Canada
bmitra@microsoft.com

Abstract

The effective training and evaluation of retrieval systems require a substantial amount of relevance judgments, which are traditionally collected from human assessors – a process that is both costly and time-consuming. Large Language Models (LLMs) have shown promise in generating relevance labels for search tasks, offering a potential alternative to manual assessments. Current approaches often rely on a single LLM, such as GPT-4, which, despite being effective, are expensive and prone to intra-model biases that can favour systems leveraging similar models. In this work, we introduce **JudgeBlender**, a framework that employs smaller, open-source models to provide relevance judgments by combining evaluations across multiple LLMs (*LLMBlender*) or multiple prompts (*PromptBlender*). By leveraging the LLMJudge benchmark [10], we compare JudgeBlender with state-of-the-art methods and the top performers in the LLMJudge challenge. Our results show that JudgeBlender achieves competitive performance, demonstrating that very large models are often unnecessary for reliable relevance assessments.

CCS Concepts

• **Information systems** → **Information retrieval**.

Keywords

LLM-as-a-Judge, Relevance Judgement, Evaluation

ACM Reference Format:
Hossein A. Rahmani, Emine Yilmaz, Nick Craswell, and Bhaskar Mitra. 2025. JudgeBlender: Ensembling Automatic Relevance Judgments. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715536

1 Introduction

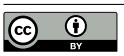
In Information Retrieval (IR), large-scale datasets that capture the relevance of documents to users' queries are critical for training and evaluating retrieval systems. Traditionally, relevance judgments have been obtained either manually, through human assessors, or via heuristic-based methods. While effective, these approaches face limitations such as scalability issues, human error, and subjective biases that can skew evaluations. With the rapid advancements in Large Language Models (LLMs), there is a growing opportunity to automate the relevance judgment process by leveraging their capabilities to comprehend and reason over large volumes of documents and passages.

LLMs have demonstrated impressive performance across various natural language processing tasks, including text classification, summarisation, and complex reasoning. However, when applied to the task of generating relevance judgments in IR, individual models may exhibit inherent biases, performance inconsistencies across domains, and susceptibility to overfitting specific linguistic patterns. These challenges make it difficult to rely solely on a single LLM for reliable relevance scoring across diverse datasets and query types.

To address these issues, we propose a novel framework, JudgeBlender, that employs an ensemble of LLMs to generate more robust and accurate relevance judgments. Instead of depending on a single model as a "judge," our approach introduces a panel of diverse evaluators, each contributing unique perspectives to the relevance evaluation process. By aggregating their outputs, we aim to achieve a more balanced and comprehensive assessment. This "jury" of models produces multiple relevance scores for each query-document pair, which are subsequently aggregated using various ensemble strategies – such as averaging, weighted voting, and advanced statistical methods – to yield a final, more reliable relevance score.

The key advantage of this approach lies in its ability to leverage the strengths of different LLMs while minimising their individual weaknesses. For instance, some models may excel at identifying semantic similarities in short texts, whereas others are better suited to processing longer, more complex documents. By ensembling models, we harness their complementary strengths, resulting in a more consistent and accurate determination of relevance.

Our contributions in this work are threefold. First, we develop a methodology for ensembling LLMs tailored to the task of generating relevance judgments. Second, we design and implement multiple aggregator functions to combine individual model outputs in ways that optimise the final relevance score. Third, we conduct extensive experiments on the LLMJudge challenge dataset [10], demonstrating that our ensemble-based approach outperforms individual LLMs by achieving higher precision and consistency in relevance assessments.



WWW Companion '25, April 28-May 2, 2025, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/25/04
https://doi.org/10.1145/3701716.3715536

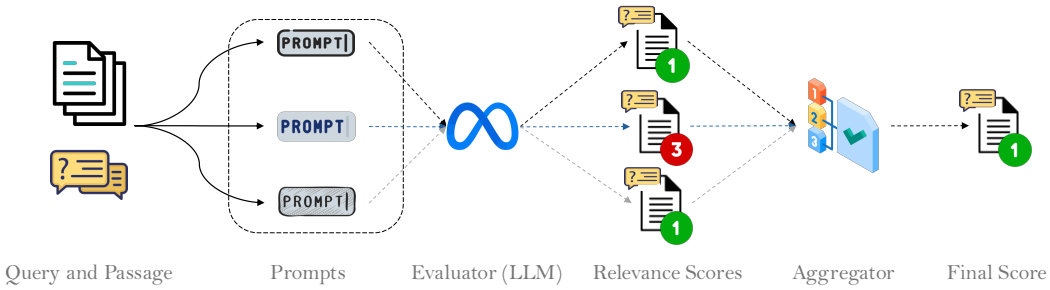


Figure 1: PromptBlender Method

2 Related Work

The high cost and time required for manual relevance judgments have motivated automated methods as scalable alternatives [3]. Large Language Models (LLMs) have shown promise in this area, with researchers exploring various prompting techniques such as zero-shot, one-shot, and few-shot learning to align LLM outputs with human evaluations [3, 13]. Techniques incorporating detailed instructions, query intent analysis, and multiple evaluators have improved alignment [13, 14]. For example, Upadhyay et al. [14] introduced UMBRELA, a structured prompting framework using GPT-4o, while Farzi and Dietz [5] developed the RUBRIC metric for query-specific evaluations. Despite these advances, reliance on commercial models introduces challenges such as high costs, reproducibility issues, and data leakage risks.

Ensemble methods have been widely adopted in machine learning to improve prediction reliability by leveraging the diversity of multiple models [2, 11]. Recent works have extended this idea to LLMs for tasks like summarisation [6], translation, and question answering [15]. For instance, Jiang et al. [6] combined LLM outputs through pairwise comparisons, while Verga et al. [15] proposed an LLM panel for evaluating free-form text generation. However, no prior work has investigated LLM ensembles for automated relevance judgments in information retrieval (IR). Our work uniquely combines diverse prompt strategies (PromptBlender) and model scores (LLMBlender) to mitigate individual model biases and enhance the precision and contextual understanding of relevance judgments.

3 JudgeBlender

JudgeBlender uses an ensemble of models to aggregate their outputs into a final result. It has two variants: PromptBlender and LLMBlender.

PromptBlender. This approach uses a single model with various prompts to evaluate the relevance of a query to a passage. The goal is to capture different perspectives by eliciting diverse reasoning pathways. The outputs are aggregated, usually by averaging or voting, to provide a multi-faceted relevance assessment from one model.

LLMBlender. LLMBlender extends this approach by using multiple models, each with a unique prompt for relevance assessment. The models are aggregated similarly to PromptBlender, leveraging their complementary strengths for a more reliable final score.

Table 1: Statistics of LLMJudge challenge dataset

	#queries	#passage	#qrels	irrelevant	related	high. rel	perfect. rel
Dev	25	7,224	7,263	4,538	1,403	625	697
Test	25	4,414	4,423	2,005	1,233	808	377

To calculate the final relevance score, the individual scores are pooled together through an aggregator function such that the final score = $f(j \in P : j(a))$ where $P \in \{PromptBlender, LLMBlender\}$ is a panel composed of individual judges j and f is an aggregator function.

4 Experimental Setup

Dataset. In our experiments, we used the LLMJudge challenge dataset [10], which is based on the TREC DL 2023 [1, 8] passage ranking task with human judgments (also known as *qrels*). A TREC DL judgment includes the query, passage, and a relevance label assigned to the passage by human experts. Relevance scores are on a four-point scale: perfectly relevant (3), highly relevant (2), related (1), and irrelevant (0). Table 1 shows the detailed statistics of the LLMJudge challenge dataset.

Aggregator Function. We explore two different aggregation functions to combine scores from multiple judges: (i) majority voting (MV) and (ii) average voting (AV). For majority voting, in the case of a tie, we apply four strategies to resolve the conflict: (1) selecting a random score (Rnd), (2) choosing the maximum score (Max), (3) choosing the minimum score (Min), or (4) taking the average (Avg) of the tied scores. This allows us to examine how different tie-breaking approaches impact the final relevance judgment. For average voting, we directly compute the average of the judges' scores as the final relevance score. Future work could explore using another LLM to break ties or make the final decision.

Models and Prompt Families. We use open-source, small language models for both variants of JudgeBlender. For the *PromptBlender* method, we employ `Meta-Llama-3-8B` as our base model and adopt three distinct prompting strategies to generate the pool of judgments: (1) the prompt proposed by Thomas et al. [13], (2) a prompt that breaks down the concept of "relevance" into multiple criteria inspired by [4], and (3) a two-step prompt that first asks for binary relevance judgment, followed by generating final scores for relevant passages.

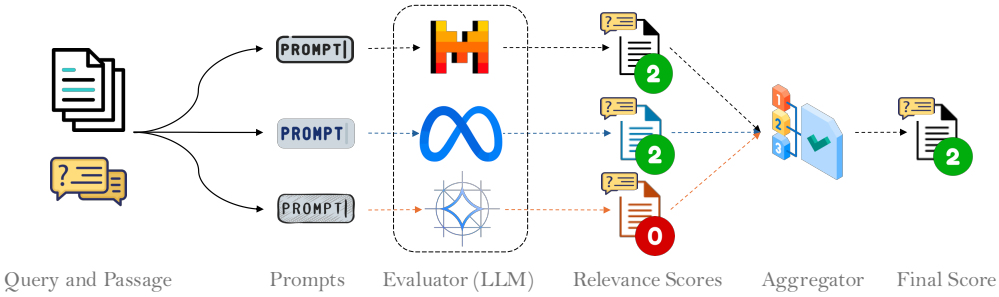


Figure 2: LLMBlender Method

For the *LLMBlender* method, judgments are derived from three distinct models belonging to different model families. Specifically, we consider `Mistral-7B`, `Gemma-7B`, and `Llama-3-8B` in our experiments. Each model is prompted using a different variant: (1) For `Mistral-7B`, we use the prompt that decomposes "relevance" into multiple criteria. For `Gemma-7B`, we employ the prompt from Thomas et al. [13], and (3) for `Llama-3-8B`, we apply the two-step prompt, which first asks for binary relevance judgment and then generates final scores for relevant passages. The codes and prompts are available on our GitHub.[1]

Evaluation Measurement. Following previous studies [3, 8, 13], we evaluate the proposed methods and baseline approaches using Cohen's κ and Krippendorff's α at the label correlation level and Spearman's ρ and Kendall's τ for the system ranking correlation.

Comparison Methods. We compare the variations of JudgeBlender including PromptBlender and LLMBlender as well as recent state-of-the-art automatic relevance judgment baselines:

- **Faggioli et al** [3]: We used the updated instructions from Figure 2 [3] by adding the definition of the relevance scores from TREC DL 2023 [1] and asking LLMs to generate the relevance score instead of binary classification.
- **Thomas et al** [13]: We used the general relevance direct grading prompt which includes the role feature (see Figure 2 [13]).
- **MultiCriteria** [4]: This is the best-performing method from the LLMJudge challenge [10]. This method evaluates the relevance of a passage to a query by breaking down the concept of "relevance" into four criteria: Exactness, Coverage, Topicality, and Contextual Fit. Each criterion is individually assessed through prompting an LLM, and the resulting scores are prompted for the final relevance judgment score.
- **Rubric (Qs.)** [5]: This method assesses relevance by evaluating how well a passage answers about 10 open-ended questions, assigning grades to each. The final relevance label is determined using a heuristic mapping of grades to relevance scores.
- **GenRE** [7]: This method fine-tuned Llama model under two different settings for 5 epoch. **GenRE-dev**: Fine-tune Llama-3-8B on the dev set of LLMJudge dataset [10]. **GenRE-trec**: Fine-tune Llama-3-8B on the qrels of TREC-DL 2019, 2020, and 2021, as well as the dev set of the LLMJudge dataset [10].

[1]https://github.com/rahmanidashti/JudgeBlender

Table 2: Judgment and system ranking correlation of JudgeBlender methods compared to JudgeBlender variation, direct LLM relevance label prompts, fine-tuned methods, and methods based on GPT-4o. κ: Cohen's Kappa, α: Krippendorff's alpha, τ: Kendall's Tau, ρ: Spearman's rank correlation. Best results per column denoted in green-bold, best across baselines methods denoted in cyan, and best per aggregator function for each variation of JudgeBlender is denoted in yellow.

Method	Model	κ	α	τ		ρ	
				NDCG@10	MAP	NDCG@10	MAP
Baselines							
Faggioli et al. [3]	GPT-35-turbo	0.0754	0.2808	0.9181	0.9054	0.9863	0.9798
Faggioli et al. [3]	GPT-4-32k	0.211	0.4642	0.9052	0.8796	0.981	0.9698
Thomas et al. [13]	GPT-35-turbo	0.1236	0.3207	0.8664	0.8968	0.9689	0.9798
Thomas et al. [13]	GPT-4-32k	0.2293	0.4877	0.9181	0.9011	0.9867	0.9778
MultiCriteria [4]	Llama-3-8B	0.1829	0.2888	0.9483	0.9140	0.9919	0.9794
Rubric (Qs.) [5]	GPT-3.5	0.0779	0.1036	0.8276	0.8839	0.9544	0.9714
Fine-tuned Methods							
GenRE-dev [7]	Llama-3-8B	0.1823	0.4069	0.9042	**0.9312**	0.9826	**0.9879**
GenRE-trec [7]	Llama-3-8B	0.1471	0.1623	0.8568	0.9011	0.9608	0.9806
Methods based on GPT-4o							
SunMulti [12]	GPT-4o	0.2388	0.4108	0.8966	0.8968	0.9798	0.977
RelExp	GPT-4o	0.2519	0.4701	0.9009	0.9140	0.9819	0.9847
PromptBlender							
PromptBlender_1		0.0465	0.1192	0.9042	0.8882	0.9822	0.9762
PromptBlender_2	Llama-3-8B	0.1741	0.3579	0.9128	0.8827	0.9838	0.9745
PromptBlender_3		0.2374	0.4482	0.9136	0.8764	0.9626	0.9649
PromptBlender							
+ MV(Avg.)		0.2398	0.4769	0.9526	0.8968	0.9919	0.9762
+ MV(Rnd.)		0.2436	0.4747	0.931	0.8925	0.9875	0.9790
+ MV(Max.)	Llama-3-8B	0.2219	0.4527	0.9085	0.8968	0.9838	0.9762
+ MV(Min.)		0.2023	0.4052	0.9085	0.8839	0.9813	0.9758
+ AV		0.2379	**0.4887**	0.9224	0.9054	0.9863	0.9798
LLMBlender							
LLMBlender_1	Mistral 7B	0.0832	0.1110	0.9267	0.8968	0.9867	0.977
LLMBlender_2	Gemma 7B	0.1880	0.3821	0.9440	0.9069	0.9907	0.9542
LLMBlender_3	Llama-3-8B	0.2454	0.4673	0.9353	0.8968	0.9883	0.9786
LLMBlender							
+ MV(Avg.)		0.2553	0.4784	**0.9612**	0.9011	**0.9940**	0.9806
+ MV(Rnd.)	Mistral 7B	**0.2619**	0.4772	0.9569	0.9011	**0.9940**	0.9810
+ MV(Max.)	Gemma 7B	0.2543	0.4620	0.9397	0.9011	0.9899	0.9806
+ MV(Min.)	Llama-3-8B	0.2600	0.4709	0.9483	0.9011	0.9923	0.9810
+ AV		0.2502	0.4832	0.9569	0.9054	0.9935	0.9815

- **SunMulti**: This method applies approach proposed by Sun et al. [12] to give a binary relevance judgment, and then generates relevance scores (1-3) only for passages marked as relevant.
- **RelExp**: This method includes reasoning for the relevance judgment within the prompt prposed by Thomas et al. [13], asking

Table 3: Judgments inter-annotator agreement on LLM-Judge challenge dataset. Comparing best variants of PromptBlender and LLMBlender to best baseline methods in terms of Cohen's κ and NDCG@10 (i.e., MultiCriteria and RelExp).

TREC	MultiCriteria				RelExp			
	0	1	2	3	0	1	2	3
3	10	26	243	98	61	126	68	122
2	43	72	596	97	206	307	186	109
1	191	244	682	116	618	417	136	62
0	783	409	692	121	1550	360	66	29
	PromptBlender - MV(Avg.)				**LLMBlender - MV(Avg.)**			
	0	1	2	3	0	1	2	3
3	45	24	271	37	25	47	204	101
2	150	112	510	36	75	179	479	75
0	583	157	456	37	393	351	410	79
1	1454	162	375	14	1189	429	340	47

the LLM to explain its assessment while rating the passage based on specific criteria.

5 Results

Correlation to Human Judgments. Table 2 presents the correlation between scores produced by different evaluator methods and human judgments, measured using Cohen's κ and Krippendorff's α. Overall, both variants of JudgeBlender achieve the strongest correlation across both metrics. In contrast, methods based on GPT-4 and fine-tuned approaches show weaker performance, particularly when evaluated using Cohen's κ.

System Ranking Correlation. Table 2 also shows how system rankings produced by different evaluation methods correlate with human judgments on TREC DL 2023 submissions. We report Kendall's τ and Spearman's rank correlation (ρ) between the system rankings generated by each method and those based on human judgments. We observe that LLMBlender achieves the highest correlation with human rankings when NDCG@10 is used as the evaluation metric. However, the results for MAP reveal slight differences: the fine-tuned method on the development set of LLMJudge and GPT-4o, when prompted to provide explanations during judgment, demonstrate better performance.

Inter-Judge Agreement Analysis. We analyze the agreement between the manual TREC DL 2023 judgments (LLMJudge challenge dataset) and the predicted relevance labels, as presented in Table 3. The inter-annotator agreement metric measures the percentage of correctly predicted judgments at each relevance level, providing insights into how closely the model's predictions align with human judgments. In our analysis, the *MultiCriteria* method achieves the highest agreement at the very relevant level, particularly for highly and perfectly relevant passages, highlighting its strength in identifying strong relevance. Conversely, the *RelExp* method demonstrates higher agreement with irrelevant passages, suggesting it is more effective at detecting non-relevant content. However, the best-performing variant of JudgeBlender, LLMBlender - MV(Avg.), demonstrates consistent and strong agreement across all four relevance levels.

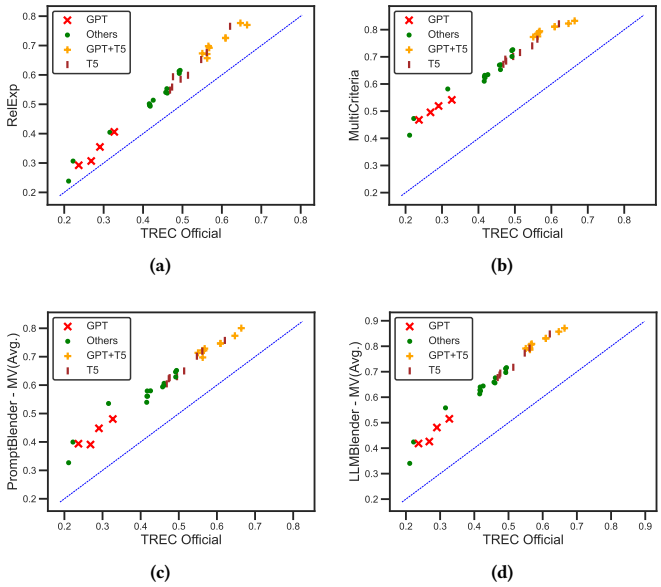


Figure 3: Scatter plots of the effectiveness of TREC Deep Learning track 2023 runs

Bias in System Evaluation. Here, we analyse the bias that evaluating on LLM-generated relevance judgment may exhibit towards systems that are based on a similar language model to the one that was used in the relevange judgment process. To do this, similar to Rahmani et al. [8], we categorised the systems submitted to TREC DL 2023 based on the approach they use (i.e., language models used in their ranking or retrieval pipeline) using the metadata file released as part of SynDL resource [9]. This results in four different system categories: systems based on **GPT**, **T5**, **GPT + T5** (i.e., a combination of GPT and T5), and **others** (i.e., traditional methods such as BM25, or any model that does not use either GPT or T5).

Figure 3 shows the effectiveness of different methods in predicting relevance judgments, compared to official human judgments (x-axis) across four methods: *RelExp*, *MultiCriteria*, *PromptBlender - MV(Avg.)*, and *LLMBlender - MV(Avg.)*, evaluated using NDCG@10. RelExp (Figure 3a) and MultiCriteria (Figure 3b) show distinct patterns. The *RelExp* method tends to overestimate for top-performing methods, GPT+T5 systems and T5-based systems. Conversely, *MultiCriteria* exhibits a noticeable overestimation of relevance for lower-performed models, comparing GPT-based systems (red crosses) in Figures 3a and 3b. The best-performing methods, PromptBlender - MV(Avg.) (Figure 3c) and LLMBlender - MV(Avg.) (Figure 3d), show more balanced performance across all system types. Both methods demonstrate a near-uniform spread along the diagonal, indicating that the blending of multiple LLM outputs reduces biases in both overestimating and underestimating relevance.

6 Conclusion and Future Work

This paper introduced JudgeBlender, showing that both PromptBlender and LLMBlender effectively evaluate LLM performance by aggregating results from diverse prompts or multiple models. These

approaches reduce costs and improve correlation with human judgments. While no single method outperforms in all settings, both consistently perform well. Although our study was limited to a small number of settings, JudgeBlender proves a robust alternative to relying on a single large model.

Future work could extend JudgeBlender by testing with more prompts, datasets, and LLMs. Optimizing panel selection and prompt strategies for cost and quality balance is an open area for research. Additionally, exploring advanced aggregation methods, such as using other LLMs for final decisions, offers potential for further study.

Acknowledgments

This work is supported by the Engineering and Physical Sciences Research Council [EP/S021566/1], the EPSRC Fellowship titled "Task Based Information Retrieval" [EP/P024289/1], the Turing Fellowship scheme.

References

[1] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2024. Overview of the TREC 2023 Deep Learning Track. In *Text REtrieval Conference (TREC)*. NIST, TREC.

[2] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*. Springer, 1–15.

[3] 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.

[4] Naghmeh Farzi and Laura Dietz. 2024. Best in Tau@ LLMJudge: Criteria-Based Relevance Evaluation with Llama3. *arXiv preprint arXiv:2410.14044* (2024).

[5] Naghmeh Farzi and Laura Dietz. 2024. Pencils down! Automatic rubric-based evaluation of retrieve/generate systems. In *Proceedings of the 2024 ACM SIGIR International Conference on Theory of Information Retrieval*. 175–184.

[6] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Comparison and Generative Fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

[7] Chuan Meng, Negar Arabzadeh, Arian Askari, Mohammad Aliannejadi, and Maarten de Rijke. 2024. Query performance prediction using relevance judgments generated by large language models. *arXiv preprint arXiv:2404.01012* (2024).

[8] Hossein A Rahmani, Nick Craswell, Emine Yilmaz, Bhaskar Mitra, and Daniel Campos. 2024. Synthetic test collections for retrieval evaluation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2647–2651.

[9] Hossein A Rahmani, Xi Wang, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, and Paul Thomas. 2024. SynDL: A Large-Scale Synthetic Test Collection for Passage Retrieval. *arXiv preprint arXiv:2408.16312* (2024).

[10] Hossein A Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles LA Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. 2024. LLMJudge: LLMs for Relevance Judgments. *arXiv preprint arXiv:2408.08896* (2024).

[11] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 8, 4 (2018), e1249.

[12] Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. *arXiv preprint arXiv:2304.09542* (2023).

[13] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large language models can accurately predict searcher preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1930–1940.

[14] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing RELevance Assessor. *arXiv preprint arXiv:2406.06519* (2024).

[15] Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv preprint arXiv:2404.18796* (2024).